Exploratory Data Analysis on AMEO data

The Aspiring Mind Employment Outcome 2015 (AMEO) dataset, released by Aspiring Minds, focuses on employment outcomes for engineering graduates.The dataset contains the employment outcomes of engineering graduates as dependent variables (Salary, Job Titles, and Job Locations) along with the standardized scores from three different areas – cognitive skills, technical skills and personality skills. The dataset also contains demographic features. It includes dependent variables such as Salary, Job Titles, and Job Locations, along with standardized scores in cognitive skills, technical skills, and personality skills. With around 40 independent variables and 4000 data points, these variables encompass both continuous and categorical data. The dataset also includes demographic features and unique identifiers for each candidate.

Features:

1. **ID:** Unique identifier for each candidate.
2. **Salary:** Annual CTC (Cost to Company) offered to the candidate (in INR).
3. **DOJ:** Date of joining the company.
4. **DOL:** Date of leaving the company.
5. **Designation:** Job title or position offered to the candidate.
6. **JobCity:** Location of the job (city).
7. **Gender:** Gender of the candidate.
8. **DOB:** Date of birth of the candidate.
9. **10percentage:** Overall marks obtained in grade 10 examinations.
10. **10board:** School board whose curriculum the candidate followed in grade 10.
11. **12graduation:** Year of graduation from senior year high school.
12. **12percentage:** Overall marks obtained in grade 12 examinations.
13. **12board:** School board whose curriculum the candidate followed in grade 12.
14. **CollegeID:** Unique identifier for the college attended by the candidate.
15. **CollegeTier:** Tier of the college attended.
16. **Degree:** Degree obtained or pursued by the candidate.
17. **Specialization:** Field of specialization pursued by the candidate.
18. **CollegeGPA:** Aggregate GPA (Grade Point Average) at graduation.
19. **CollegeCityID:** Unique identifier for the city where the college is located.
20. **CollegeCityTier:** Tier of the city where the college is located.
21. **CollegeState:** Name of the state where the college is located.
22. **GraduationYear:** Year of graduation with a bachelor's degree.
23. **English:** Scores in AMCAT English section.
24. **Logical:** Scores in AMCAT Logical section.
25. **Quant:** Scores in AMCAT Quantitative section.
26. **Domain:** Scores in AMCAT's domain module.
27. **ComputerProgramming:** Score in AMCAT's Computer programming section.
28. **ElectronicsAndSemicon:** Score in AMCAT's Electronics & Semiconductor Engineering section.
29. **ComputerScience:** Score in AMCAT's Computer Science section.

30. **MechanicalEngg:** Score in AMCAT's Mechanical Engineering section.
31. **ElectricalEngg:** Score in AMCAT's Electrical Engineering section.
32. **TelecomEngg:** Score in AMCAT's Telecommunication Engineering section.
33. **CivilEngg:** Score in AMCAT's Civil Engineering section.
34. **Conscientiousness:** Scores in one of the sections of AMCAT's personality test.
35. **Agreeableness:** Scores in one of the sections of AMCAT's personality test.
36. **Extraversion:** Scores in one of the sections of AMCAT's personality test.
37. **Neuroticism:** Scores in one of the sections of AMCAT's personality test.
38. **Openness_to_experience:** Scores in one of the sections of AMCAT's personality test.

Objective:

The objective of this Exploratory Data Analysis (EDA) is to comprehensively explore the Aspiring Mind Employment Outcome 2015 (AMEO) dataset, focusing on understanding the relationship between various features and the target variable, Salary.

Key Goals:

1. **Dataset Overview:** Provide a detailed overview of the dataset's features, including continuous and categorical variables, as well as demographic information and unique identifiers.
2. **Pattern Identification:** Identify any discernible patterns or trends within the dataset through visualizations and statistical analysis.
3. **Relationship Investigation:** Investigate the relationships between independent variables and the target variable (Salary) to uncover insights into factors influencing salary outcomes.
4. **Outlier Detection:** Perform outlier detection to identify any anomalies or abnormalities in the dataset that may affect the analysis.
5. **Insights and Recommendations:** Offer practical insights and recommendations based on the analysis findings, aiming to provide valuable insights for stakeholders and decision-makers.

IMPORT LIBRARIES

In [1]:
```
import pandas as pd
```

```
C:\Users\prasa\AppData\Local\Temp\ipykernel_15216\4080736814.py:1: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other lib
raries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
import seaborn as sns
# import warnings
import warnings
warnings.filterwarnings('ignore')
```

Import the CSV File

In [5]:
```
df=pd.read_csv(r"C:\Users\prasa\Downloads\data.xlsx - Sheet1.csv")
```

In [6]:
```
df.head()
```

Out[6]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	ComputerScience	MechanicalEng
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	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	...	-1	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	...	-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	
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	

5 rows × 39 columns

In [7]:
```
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
 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
```

Descriptive statistics of Numerical Columns

In [8]:
```
df.describe()
```

Out[8]:

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID	Colleg
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	399
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	5156.851426	1.925713	71.486171	5156.851426	
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	4802.261482	0.262270	8.167338	4802.261482	
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	2.000000	1.000000	6.450000	2.000000	
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	494.000000	2.000000	66.407500	494.000000	
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	3879.000000	2.000000	71.720000	3879.000000	
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	8818.000000	2.000000	76.327500	8818.000000	
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	18409.000000	2.000000	99.930000	18409.000000	

8 rows × 27 columns

Unique Values in Each column

In [9]:
```
df.nunique()
```

Out[9]:
```
Unnamed: 0                1
ID                     3998
Salary                  177
DOJ                      81
DOL                      67
Designation             419
JobCity                 339
Gender                    2
DOB                    1872
10percentage            851
10board                 275
12graduation             16
12percentage            801
12board                 340
CollegeID              1350
CollegeTier               2
Degree                    4
Specialization           46
collegeGPA             1282
CollegeCityID          1350
CollegeCityTier           2
CollegeState             26
GraduationYear           11
English                 111
Logical                 107
Quant                   138
Domain                  243
ComputerProgramming      79
ElectronicsAndSemicon    29
ComputerScience          20
MechanicalEngg           42
ElectricalEngg           31
TelecomEngg              26
CivilEngg                23
conscientiousness       141
agreeableness           149
extraversion            154
nueroticism             217
openess_to_experience   142
dtype: int64
```

Checking for Null Values

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Unnamed: 0               0
ID                       0
Salary                   0
DOJ                      0
DOL                      0
Designation              0
JobCity                  0
Gender                   0
DOB                      0
10percentage             0
10board                  0
12graduation             0
12percentage             0
12board                  0
CollegeID                0
CollegeTier              0
Degree                   0
Specialization           0
collegeGPA               0
CollegeCityID            0
CollegeCityTier          0
CollegeState             0
GraduationYear           0
English                  0
Logical                  0
Quant                    0
Domain                   0
ComputerProgramming      0
ElectronicsAndSemicon    0
ComputerScience          0
MechanicalEngg           0
ElectricalEngg           0
TelecomEngg              0
CivilEngg                0
conscientiousness        0
agreeableness            0
extraversion             0
nueroticism              0
openess_to_experience    0
dtype: int64
```

Dropping Unwanted Column

In [11]:
```
df.drop("Unnamed: 0",axis=1,inplace=True)
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['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')
```

Changing "Present" to Date and Changing Datatype of Date Columns

In [20]:
```
df['DOL'].replace('present','31-12-2015', inplace = True)
# Convert 'DOJ' and 'DOL' to datetime
df['DOJ'] = pd.to_datetime(df['DOJ'])
df['DOL'] = pd.to_datetime(df['DOL'],format="mixed",dayfirst=True)

# Convert 'DOB' to datetimeA
df['DOB'] = pd.to_datetime(df['DOB'])

df['Designation'] = df['Designation'].astype('category')
df['JobCity'] = df['JobCity'].astype('category')
df['Gender'] = df['Gender'].astype('category')
df['CollegeCityTier'] = df['CollegeCityTier'].astype('category')

# Print the data types after conversion
print(df.dtypes)
```

```
ID                             int64
Salary                       float64
DOJ                   datetime64[ns]
DOL                   datetime64[ns]
Designation                 category
JobCity                     category
Gender                      category
DOB                   datetime64[ns]
10percentage                 float64
10board                       object
12graduation                   int64
12percentage                 float64
12board                       object
CollegeID                      int64
CollegeTier                    int64
Degree                        object
Specialization                object
collegeGPA                   float64
CollegeCityID                  int64
CollegeCityTier             category
CollegeState                  object
GraduationYear                 int64
English                        int64
Logical                        int64
Quant                          int64
Domain                       float64
ComputerProgramming            int64
ElectronicsAndSemicon          int64
ComputerScience                int64
MechanicalEngg                 int64
ElectricalEngg                 int64
TelecomEngg                    int64
CivilEngg                      int64
conscientiousness            float64
agreeableness                float64
extraversion                 float64
nueroticism                  float64
openess_to_experience        float64
dtype: object
```

Categorising the Designations to four Categories

In [21]:

```
designation_mapping = {
    'Programming Analyst': [
        'Programmer Analyst', 'Programmer Analyst Trainee', 'Program Analyst Trainee',
        'Software Engineer Analyst', 'Technical Analyst', 'Software Programmer',
        'Junior Software Developer', 'Software Tester', 'Database Developer',
        'Software Test Engineer (ETL)', 'Associate QA', 'QA Trainee',
        'Program Analyst Trainee', 'Software Test Engineerte', 'Help Desk Technician',
        'Help Desk Analyst', 'Website Developer/Tester', 'Database Developer',
        'Web Developer', 'SEO Analyst', 'Software Designer', 'SEO Trainee',
        'SEO Executive', 'Digital Marketing Specialist', 'Game Developer',
        'Full Stack Developer', 'Front End Developer', 'Ruby on Rails Developer',
        'Web Intern', 'Junior .NET Developer', 'Dotnet Developer',
        'Web Designer and Joomla Administrator', 'Website Developer/Tester',
        'Java Trainee', 'Assistant System Engineer - Trainee', 'Research Scientist',
        'Technical Writer', 'Technical Consultant', 'Business Systems Analyst',
        'Business System Analyst', 'Business Analyst Consultant', 'Business Consultant',
        'Systems Analyst', 'Management Trainee', 'Entry Level Management Trainee',
        'Graduate Engineer Trainee', 'Graduate Apprentice Trainee', 'Junior Manager',
        'Lecturer & Electrical Maintenance', 'Visiting Faculty', 'Assistant Professor',
        'Professor', 'Educator', 'Research Engineer', 'Research Analyst',
        'Research Associate', 'Quality Assurance Analyst', 'Quality Assurance Test Engineer',
        'Quality Assurance Automation Engineer', 'Quality Assurance Tester',
        'Quality Assurance Auditor', 'Quality Engineer', 'Quality Control Engineer',
        'Quality Control Inspection Technician', 'Quality Controller',
        'Quality Consultant', 'MIS Executive', 'Process Executive', 'Process Control Engineer',
        'Process Advisor', 'Technical Recruiter', 'Recruitment Coordinator',
        'Staffing Recruiter', 'Corporate Recruiter', 'Talent Acquisition Specialist',
        'HR Recruiter', 'Executive Recruiter', 'Recruiter', 'Delivery Software Engineer',
        'Software Test Engineer', 'Senior Quality Engineer', 'QA Analyst',
        'Product Development Engineer', 'Data Entry Operator', 'Developer', 'ASE',
        'Telecommunication Engineer', 'Executive Assistant', 'Online Marketing Manager',
        'Documentation Specialist', 'Site Manager', 'Production Engineer',
        'Customer Service', 'Test Engineer', 'Java Developer', 'Engineer',
        'Data Analyst', 'Faculty', 'Customer Service Representative',
        'PHP Developer', 'Programmer', 'Business Analyst', 'Application Engineer',
        'SAP Consultant', 'Quality Analyst', 'Marketing Coordinator',
        'Senior Engineer', 'Business Development Managerde', 'Network Engineer',
        'Network Administrator', 'Business Development Executive',
        'Trainee Software Engineer', 'Design Engineer', 'Sales Associate',
        'Sales Engineer', 'Human Resources Associate', 'Mobile Application Developer',
        'IT Support Specialist', 'Business Process Analyst', 'Marketing Assistant',
        'Sales Executive', 'Admin Assistant', 'Account Executive', 'Oracle DBA',
        'Lecturer', '.NET Web Developer', 'Marketing Executive',
        'Client Services Associate', 'IT Analyst', 'Senior Developer',
        'CAD Designer', 'Business Technology Analyst', 'Asst. Manager',
```

```
        'Service Engineer', 'Web Designer', 'Software Quality Assurance Tester',
        'Software Quality Assurance Analyst', 'Designer', 'Business Systems Consultant',
        'Business Development Manager', 'Linux Systems Administrator',
        'Software Trainee', 'Entry Level Sales and Marketing',
        'Electrical Field Engineer', 'Windows Systems Administrator',
        'Python Developer', 'Web Application Developer', 'Assistant Systems Engineer',
        'JavaScript Developer', 'Performance Engineer', 'Portfolio Analyst',
        'Associate System Engineer', 'Mechanical Design Engineer', 'Product Engineer',
        'Network Security Engineer', 'Operations', 'Automation Engineer',
        'Data Scientist', 'Quality Associate', 'Manual Tester', 'Sr. Engineer',
        'Telecom Support Engineer', 'Engineer- Customer Support', 'Branch Manager',
        'Technology Lead', 'Software Trainee Engineer', 'DCS Engineer',
        'UX Designer', 'Clerical', 'HR Generalist', 'Database Administrator',
        'Senior Design Engineer', 'SEO', 'Marketing Analyst', 'IT Executive',
        'Salesforce Developer', 'SQL DBA', 'Manager', 'Senior Business Analyst',
        'C# Developer', 'Implementation Engineer', 'Executive HR',
        'Executive Engineer', 'SharePoint Developer', 'Sales Management Trainee',
        'Senior Project Engineer', 'IT Recruiter', 'Desktop Support Technician',
        'Continuous Improvement Engineer', 'ETL Developer',
        'Sales and Service Engineer', 'Project Manager', 'Training Specialist',
        'Product Manager', 'Assistant Programmer', 'Principal Software Engineer',
        'Software Devloper', 'Senior Mechanical Engineer', 'Service Coordinator',
        'Testing Engineer', 'SAP ABAP Consultant', 'SEO Engineer',
        'Sales Account Manager', 'Software Engineer Trainee',
        'Customer Service Manager', 'General Manager', 'Engineering Manager',
        'Senior Network Engineer', 'Field Based Employee Relations Manager',
        'Phone Banking Officer', 'Support Engineer', 'Associate Test Engineer',
        'Technology Analyst', 'Network Support Engineer', 'IT Business Analyst',
        'Junior System Analyst', 'Senior .NET Developer', 'Secretary',
        'Office Coordinator', 'HR Manager', 'HTML Developer', 'Sales Support',
        'Front End Web Developer', 'Administrative Support', 'Territory Sales Manager',
        'Environmental Engineer', 'Web Designer and SEO', 'Information Security Analyst',
        'Field Business Development Associate', 'Operational Executive',
        'Administrative Coordinator', 'Senior Risk Consultant', 'CAD Drafter',
        'NOC Engineer', 'Human Resources Intern', 'Senior Quality Assurance Engineer',
        'Clerical Assistant', 'Software Enginner', 'Quality Assurance',
        'Graphic Designer', 'Sales Development Manager',
        'Business Intelligence Analyst', 'Team Lead', 'Operational Excellence Manager',
        'Sales & Service Engineer', 'Sr. Database Engineer', 'Software Engineer Associate',
        'Executive Engg', 'IT Technician', 'Business Office Manager',
        'Quality Control Inspector', 'Product Design Engineer', 'Manufacturing Engineer',
        'SAP Analyst', 'Software Engineere', 'Financial Service Consultant',
        'Co Faculty', 'Software Analyst', 'Desktop Support Analyst',
        'Graduate Engineer', 'Engineering Technician', 'IT Assistant',
        'Marketing Manager', 'Human Resource Assistant', 'HR Assistant',
        'Product Developer', 'Customer Support Engineer', 'GIS/CAD Engineer',
        'Senior Web Developer', 'SQL Developer', 'Research Staff Member',
        'SAP ABAP Associate Consultant', 'Project Management Officer',
        'Sales Manager', 'Catalog Associate', 'Assistant Store Manager',
        'Software Engg', 'IT Developer', 'Apprentice', 'Risk Consultant',
        'Account Manager', 'Assistant Administrator', 'Civil Engineer',
        'Service Manager', 'Teradata DBA', 'Full-Time Loss Prevention Associate',
        'Junior Recruiter', 'Associate Developer', 'Shift Engineer',
        'RF/DT Engineer', 'Human Resources Analyst', 'R&D Engineer', 'Engineer-HWS',
        'Senior Research Fellow', 'Enterprise Solutions Developer',
        'Information Technology Specialist', 'CNC Programmer', 'Financial Analyst',
        'System Engineer Trainee', 'SAP MM Consultant', 'Teradata Developer',
        'HR Executive', 'Senior Programmer', 'Associate Software Engg',
        'Supply Chain Analyst', 'Sales Trainer', 'Software Executive',
        'Team Leader', 'Risk Investigator', 'Executive Administrative Assistant',
        'Program Manager', 'R & D', 'SAP Functional Consultant',
        'Sales Coordinator', 'QA Engineer', 'Aircraft Technician',
        'Customer Care Executive', 'Senior Test Engineer',
        'Electrical Controls Engineer', 'Editor', 'BSS Engineer', 'DBA',
        'Software Eng', 'Computer Faculty', 'Recruitment Associate',
        'Logistics Executive', 'Senior Sales Executive', 'DB2 DBA',
        'Test Technician', 'Software Engineering Associate', 'Jr. Software Developer'
    ],
    'Software Engineer': [
        'Senior Software Engineer', 'Java Software Engineer', 'Senior PHP Developer',
        'Senior Systems Engineer', 'Systems Engineer', 'System Engineer',
        'Software Engineer', 'Associate Software Developer', 'Software Developer',
        '.NET Developer', 'Jr. Software Engineer', 'Trainee Software Developer',
        'Assistant System Engineer', 'Android Developer', 'iOS Developer',
        'Application Developer', 'UI Developer', 'Software Architect',
        'Senior Java Developer', 'ASP.NET Developer', 'Web Developer',
        'Full Stack Developer', 'Front End Developer', 'Software Development Engineer',
        'Senior Software Developer', 'Lead Engineer', 'Associate Technical Operations',
        'Operation Engineer', 'Operation Executive', 'Operations Engineer and Jetty Handling',
        'Operations Manager', 'Operations Analyst', 'Operations Assistant',
        'Operations Executive', 'Operations Engineer', 'Systems Administrator',
        'System Administrator', 'System Analyst', 'Systems Analyst', 'Technical Lead',
        'Technical Support Engineer', 'Technical Support Specialist',
        'Technical Support Executive', 'Technical Operations Analyst',
        'Technical Engineer', 'Technical Writer', 'Technical Assistant'
    ],
    'Hardware Engineer': [
        'Electrical Engineer', 'Mechanical Engineer', 'Electrical Project Engineer',
        'Electrical Designer', 'Electrical Design Engineer', 'Hardware Engineer',
```

```
        'Embedded Engineer', 'RF Engineer', 'Firmware Engineer', 'Embedded Software Engineer',
        'Electronic Field Service Engineer', 'Hardware Engineer', 'Industrial Engineer',
        'Controls Engineer', 'Maintenance Engineer', 'Quality Control Inspection Technician'
    ],
    'Associate Engineer': [
        'Associate Engineer', 'Associate Software Engineer', 'Assistant Software Engineer',
        'Engineer Trainee', 'Associate Engineer', 'Associate Manager',
        'Graduate Trainee Engineer', 'Assistant System Engineer', 'Systems Engineer Trainee',
        'Graduate Apprentice Trainee', 'Assistant System Engineer Trainee', 'Trainee Engineer',
        'Trainee Decision Scientist', 'Trainee Software Developer', 'Junior Engineer',
        'Junior Software Developer', 'Junior .NET Developer', 'Junior Engineer Product Support',
        'Junior Manager', 'Junior Research Fellow', 'Junior Software Engineer',
        'Assistant Engineer', 'Assistant Electrical Engineer', 'Field Engineer',
        'Telecom Engineer', 'Field Service Engineer', 'Field Service Engineer',
        'Service and Sales Engineer', 'Service and Sales Engineer', 'Cloud Engineer',
        'Desktop Support Engineer', 'IT Engineer', 'IT Operations Associate',
        'Maintenance Supervisor', 'Site Engineer', 'Project Engineer',
        'Project Coordinator', 'Project Administrator', 'Project Assistant',
        'Planning Engineer', 'Process Engineer', 'Process Associate', 'Process Executive',
        'Process Control Engineer', 'Process Advisor'
    ]
}

# Define the function to map designations to categories
def map_designation(designation):
    for key, value in designation_mapping.items():
        if designation.lower() in [x.lower() for x in value]:
            return key
    return 'Other'  # If designation not found in mapping, assign it to 'Other' category

# Apply the mapping function to create the new column
df['Designation_category'] = df['Designation'].apply(map_designation)

# Display the updated DataFrame
print(df.columns)
```

```
Index(['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',
       'Designation_category'],
      dtype='object')
```

TOP 15 Designations

In [22]:

```
top_15_designations = df['Designation'].value_counts().head(15).index

# Replace all designations not in the top 15 with 'Others'
df['Designation'] = df['Designation'].astype('object')  # Convert to object type temporarily
df.loc[~df['Designation'].isin(top_15_designations), 'Designation'] = 'Others'

# Convert back to categorical
df['Designation'] = df['Designation'].astype('category')

# Print the updated DataFrame
print(df['Designation'])
```

```
0                        Others
1             assistant manager
2              systems engineer
3       senior software engineer
4                        Others
                  ...
3993          software engineer
3994                     Others
3995                     Others
3996         software developer
3997                     Others
Name: Designation, Length: 3998, dtype: category
Categories (16, object): ['Others', 'application developer', 'assistant manager', 'java developer', ..., 'syste
ms engineer', 'technical support engineer', 'test engineer', 'web developer']
```

Checking for 0 and -1 values

In [23]:

```
# Iterate through each column
for col in df.columns:
        zero_count = (df[col] == 0).sum()
        neg_one_count = (df[col] == -1).sum()

        # Print the column name and counts if there are any occurrences
        if zero_count > 0 or neg_one_count > 0:
            print(f"Column '{col}':")
            if zero_count > 0:
```

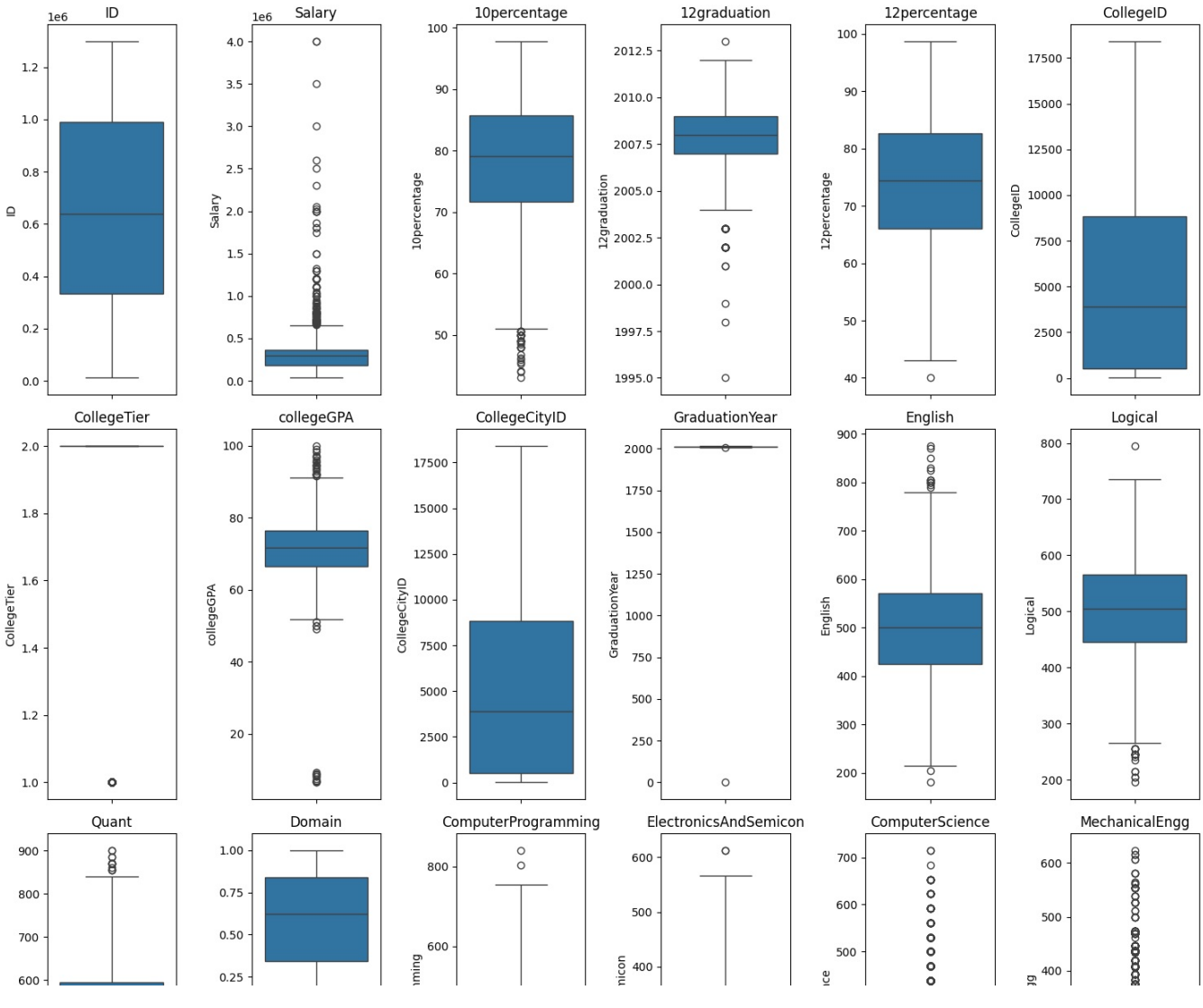
```
                print(f"   Number of 0 values: {zero_count}")
            if neg_one_count > 0:
                print(f"   Number of -1 values: {neg_one_count}")
```

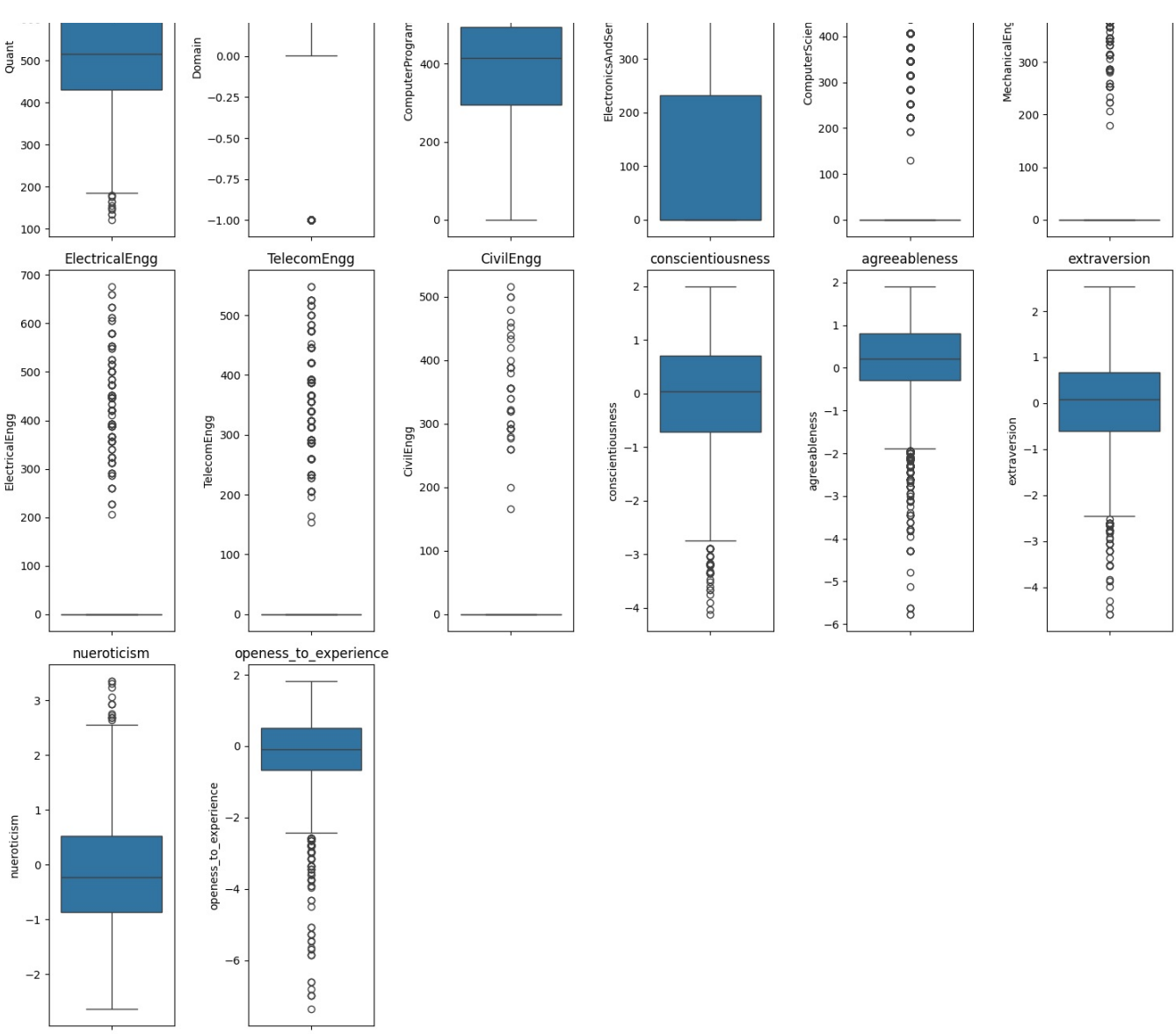
```
Column 'CollegeCityTier':
   Number of 0 values: 2797
Column 'GraduationYear':
   Number of 0 values: 1
Column 'Domain':
   Number of -1 values: 246
Column 'ComputerProgramming':
   Number of -1 values: 868
Column 'ElectronicsAndSemicon':
   Number of -1 values: 2854
Column 'ComputerScience':
   Number of -1 values: 3096
Column 'MechanicalEngg':
   Number of -1 values: 3763
Column 'ElectricalEngg':
   Number of -1 values: 3837
Column 'TelecomEngg':
   Number of -1 values: 3624
Column 'CivilEngg':
   Number of -1 values: 3956
```

Checking For Outliers

In [24]:

```
# Select numeric columns
numeric_columns = df.select_dtypes(include=['int64', 'float64']).columns

# Calculate the number of rows and columns for subplots
num_cols = 6
num_rows = (len(numeric_columns) + num_cols - 1) // num_cols

# Plot boxplots for each numeric column
plt.figure(figsize=(15, 5*num_rows))
for i, col in enumerate(numeric_columns):
    plt.subplot(num_rows, num_cols, i + 1)
    sns.boxplot(y=df[col])
    plt.title(col)
    plt.tight_layout()

plt.show()
```





Shape of DataFrame Before Data Cleaning

In [25]:
```
df.shape
```

Out[25]:
```
(3998, 39)
```

Separating Numerical and Categorical Columns

In [26]:
```
# Select numerical columns
num_cols = df.select_dtypes(include=['int64', 'float64']).columns
num_df = df[num_cols]
print(num_cols)
# Select categorical columns
cat_cols = df.select_dtypes(include=['object', 'datetime64']).columns
cat_df = df[cat_cols]
cat_df.columns
```

Out[26]:
```
Index(['ID', 'Salary', '10percentage', '12graduation', '12percentage',
       'CollegeID', 'CollegeTier', 'collegeGPA', 'CollegeCityID',
       'GraduationYear', 'English', 'Logical', 'Quant', 'Domain',
       'ComputerProgramming', 'ElectronicsAndSemicon', 'ComputerScience',
       'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg',
       'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
Index(['DOJ', 'DOL', 'DOB', '10board', '12board', 'Degree', 'Specialization',
       'CollegeState', 'Designation_category'],
      dtype='object')
```

Dropping least significant columns

In [27]:
```
columns_to_drop = ['ComputerScience','Domain', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg',
df1 = df.drop(columns=columns_to_drop)
```

In [28]:
```
df1.shape
```

Out[28]:
```
(3998, 31)
```

In [29]:
```
df1.columns
```

Out[29]:
```
Index(['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', 'conscientiousness', 'agreeableness',
       'extraversion', 'nueroticism', 'openess_to_experience',
       'Designation_category'],
      dtype='object')
```

Calculating Age From DOB column

In [30]:
```
df1['DOB'] = pd.to_datetime(df1['DOB'])
df1['Age'] = 2015 - df1['DOB'].dt.year
df1.head()
```

Out[30]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	English	Logical	Quant	conscien
0	203097	420000.0	2012-06-01	2015-12-31	Others	Bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	...	515	585	525	
1	579905	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.4	cbse	...	695	610	780	
2	810601	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.0	cbse	...	615	545	370	
3	267447	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.6	cbse	...	635	585	625	
4	343523	200000.0	2014-03-01	2015-01-03	Others	Manesar	m	1991-02-27	78.0	cbse	...	545	625	465	

5 rows × 32 columns

Removing Rows where the data is invalid

In [31]:
```
df1 = df1.drop(df1[~(df1['DOL'] > df1['DOJ'])].index)
print(df1.shape)
```

```
(3784, 32)
```

Calculating Tenure of the Candidate

In [32]:
```
data = (df1['DOL'] - df1['DOJ'])
df1['Tenure'] = (data.dt.days / 365).round(2)
```

Correcting the CGPA in CollegeGPA column

In [33]:
```
df1.loc[df1['collegeGPA']<=10,'collegeGPA'] = (df1.loc[df1['collegeGPA']<=10,'collegeGPA']/10)*100
df1.head()
```

Out[33]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Logical	Quant	conscientiousness
0	203097	420000.0	2012-06-01	2015-12-31	Others	Bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	...	585	525	0.9737
1	579905	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.4	cbse	...	610	780	-0.7335
2	810601	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.0	cbse	...	545	370	0.2718
3	267447	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.6	cbse	...	585	625	0.0464
4	343523	200000.0	2014-03-01	2015-01-03	Others	Manesar	m	1991-02-27	78.0	cbse	...	625	465	-0.8810

5 rows × 33 columns

Checking the rows which has outliers

In [34]:
```
from scipy import stats
# Select numeric columns
```

```
numeric_columns = df1.select_dtypes(include=['int64', 'float64']).columns

# Initialize an empty list to store DataFrames of outliers for each column
outliers_dfs = []

# Define a threshold for identifying outliers based on Z-score
threshold = 3

# Iterate over numeric columns to identify and store outliers
for col in numeric_columns:
    z_scores = np.abs(stats.zscore(df1[col]))
    outliers = df1[(z_scores > threshold)]
    outliers_dfs.append(outliers)

# Concatenate all DataFrames in the outliers_dfs list into a single DataFrame
outliers_df = pd.concat(outliers_dfs)

# Remove duplicate rows, if any
outliers_df = outliers_df.drop_duplicates()

# Print the DataFrame containing outliers
outliers_df
```

Out[34]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Logical	Quant	conscientiousnes
3	267447	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.60	cbse	...	585	625	0.046
123	312164	1200000.0	2010-07-01	2011-01-07	Others	Maharajganj	m	1988-04-25	59.80	icse	...	595	405	0.200
166	323894	1860000.0	2012-07-01	2014-01-10	Others	mumbai	f	1989-10-15	89.70	0	...	555	625	-1.344
383	53921	1110000.0	2010-07-01	2015-01-04	Others	Bangalore	m	1988-03-25	75.40	cbse	...	535	655	1.282
394	30706	1200000.0	2011-02-01	2015-12-31	Others	Bangalore	m	1987-10-02	89.33	icse	...	685	735	1.420
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
3139	244806	490000.0	2007-07-01	2015-12-31	Others	Bangalore	m	1989-10-30	83.20	sslc	...	415	485	-0.417
3250	453953	480000.0	2010-08-01	2015-12-31	Others	Navi mumbai	m	1985-09-24	69.33	0	...	485	285	-0.301
3459	47457	700000.0	2010-08-01	2015-12-31	Others	-1	m	1988-09-26	88.00	matric	...	375	505	0.664
3574	103947	325000.0	2010-08-01	2015-12-31	Others	Chennai	m	1989-07-13	81.40	0	...	335	345	-1.035
3717	144911	530000.0	2010-04-01	2015-12-31	Others	Faridabad	m	1987-12-31	63.00	hbsc	...	295	465	-0.301

489 rows × 33 columns

Replacing 0 and -1 values

In [35]:
```
df1['10board'] = df1['10board'].astype(str)
df1['12board'] = df1['12board'].astype(str)
df1['JobCity'] = df1['JobCity'].astype(str)
```

In [36]:
```
df1['10board'] = df1['10board'].replace({'0':np.nan})
df1['12board'] = df1['12board'].replace({'0':np.nan})
df1['GraduationYear'] = df1['GraduationYear'].replace({0:np.nan})
df1['JobCity'] = df1['JobCity'].replace({'-1':np.nan})
```

In [37]:
```
df1['10board'] = df1['10board'].astype('category')
df1['12board'] = df1['12board'].astype('category')
df1['JobCity'] = df1['JobCity'].astype('category')
```

In [38]:
```
df1['10board'].fillna(df1['10board'].mode()[0], inplace = True)
df1['12board'].fillna(df1['12board'].mode()[0], inplace = True)
df1['GraduationYear'].fillna(df1['GraduationYear'].mode()[0], inplace = True)
df1['JobCity'].fillna(df1['JobCity'].mode()[0], inplace = True)
```

In [39]:
```
# selecting numerical columns
num_cols = df1.select_dtypes(include=['int64', 'float64']).columns
num_df = df1[num_cols]
print(num_cols)

# Selecting categorical columns
cat_cols = df1.select_dtypes(include=['object', 'datetime64']).columns
cat_df = df1[cat_cols]
cat_df.columns
```

```
Index(['ID', 'Salary', '10percentage', '12graduation', '12percentage',
       'CollegeID', 'CollegeTier', 'collegeGPA', 'CollegeCityID',
       'GraduationYear', 'English', 'Logical', 'Quant', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience',
       'Tenure'],
      dtype='object')
```

Out[39]:

```
Index(['DOJ', 'DOL', 'DOB', 'Degree', 'Specialization', 'CollegeState',
       'Designation_category'],
      dtype='object')
```

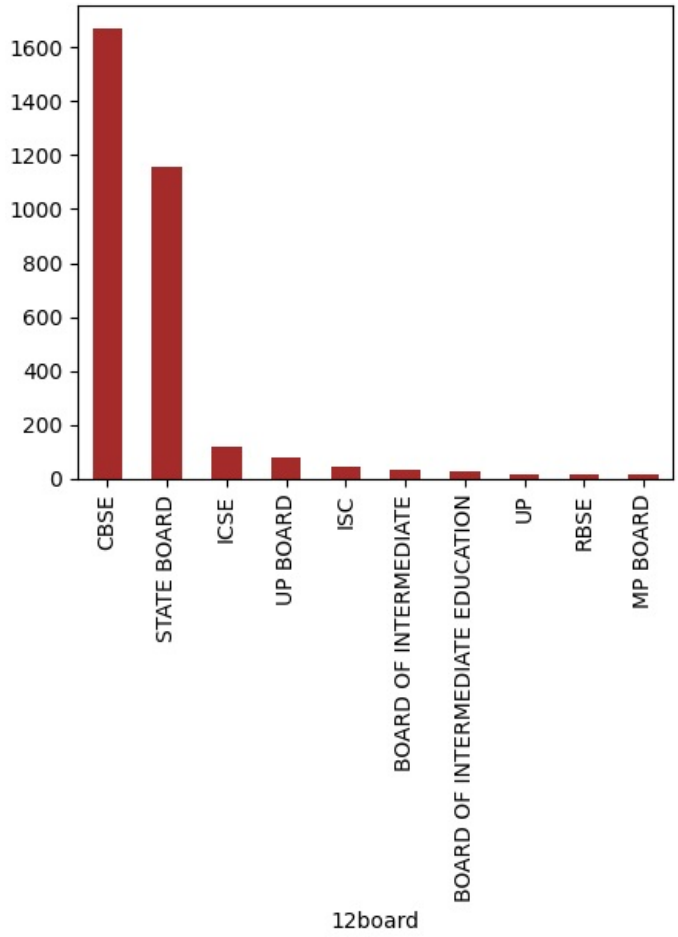
Univariate Analysis

1.1 12board

In [40]:
```
(df1['12board'].str.upper().value_counts().head(10)).plot(kind='bar',color='brown',figsize=(5, 4))
```

Out[40]:

```
<Axes: xlabel='12board'>
```



`insight`:

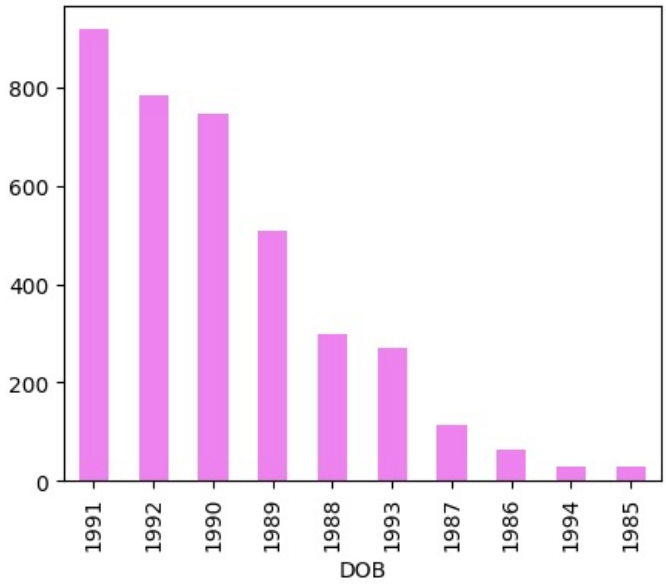
The analysis reveals that a significant portion of candidates in the employment records completed their secondary education under the curriculum of the Central Board of Secondary Education (CBSE), making it the most common educational background among the candidates. Following CBSE, candidates educated under their respective state boards represent the second-largest group. This trend suggests a prevalent preference for CBSE-affiliated schools among individuals entering the workforce, possibly influenced by the widespread recognition and adoption of CBSE curriculum nationwide. These insights have implications for workforce diversity, hiring practices, and educational policies, highlighting the importance of considering school board affiliations in employment contexts.

1.2 Date of Birth

In [41]:
```
(df1['DOB'].dt.year.value_counts().head(10)).plot(kind='bar',color='violet',figsize=(5, 4))
```

Out[41]:

```
<Axes: xlabel='DOB'>
```



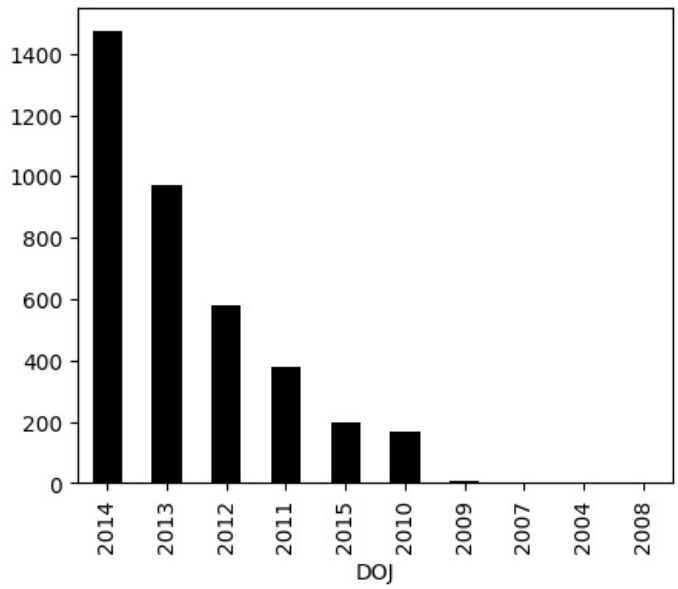
`insight` :

The analysis reveals that candidates born in the year 1991 constitute the largest group of individuals within the dataset, indicating a substantial representation from this birth year cohort. Following closely behind, candidates born in the year 1992 form the second-largest group of participants. Additionally, there is a notable presence of candidates born in the year 1985, albeit in smaller numbers compared to the aforementioned cohorts. This observation underscores the diverse age distribution among individuals in the dataset, with participants spanning different birth years actively contributing to the dataset. Such diversity in age groups highlights the broad demographic reach and inclusivity of the dataset, accommodating individuals across various stages of their academic and professional journeys.

1.3 date of joining

In [42]:
```
(df1['DOJ'].dt.year.value_counts().head(10)).plot(kind='bar',color='black',figsize=(5, 4))
```

Out[42]:
```
<Axes: xlabel='DOJ'>
```



`insight` :

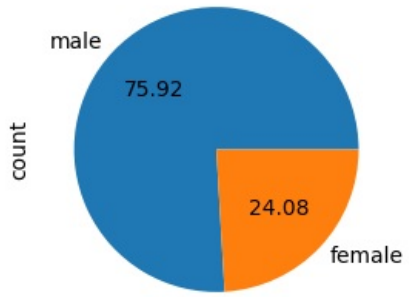
The analysis reveals a discernible trend in employment dynamics over the years, notably showing an increasing trend from 2010 to 2014. Specifically, there is a notable surge in the number of individuals joining in the year 2014, indicating a peak in employment opportunities during that period. This upward trajectory suggests a progressive growth in job opportunities or recruitment activities over the specified timeframe. Conversely, the years 2004, 2007, 2008, and 2009 exhibit either stagnant or declining trends in hiring, indicating periods of

reduced employment or limited job availability. Identifying and understanding such trends is crucial for stakeholders, as it provides valuable insights into the overall employment landscape and helps in making informed decisions regarding workforce planning and resource allocation.

1.4 Gender

In [43]:
```
labels=['male','female']
(df1['Gender'].value_counts().head(10)).plot(kind='pie',labels=labels,figsize=(3, 3),autopct='%.2f')
```

Out[43]:
```
<Axes: ylabel='count'>
```



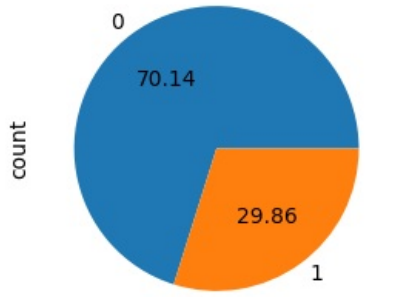
`insight` :

The analysis indicates a notable gender disparity within the employment dataset, with a significantly higher number of male employees compared to female employees. Specifically, the number of male employees is approximately three times that of female employees, highlighting a substantial gender gap in employment representation. This observation suggests that employment opportunities for females are comparatively limited or less prevalent within the dataset. Addressing such gender disparities in the workforce is essential for promoting gender equality and ensuring equitable access to employment opportunities for all individuals, regardless of gender.

1.5 Collegecitytier

In [44]:
```
(df1['CollegeCityTier'].value_counts().head(10)).plot(kind='pie',color='magenta',figsize=(3, 3),autopct="%.2f")
```

Out[44]:
```
<Axes: ylabel='count'>
```



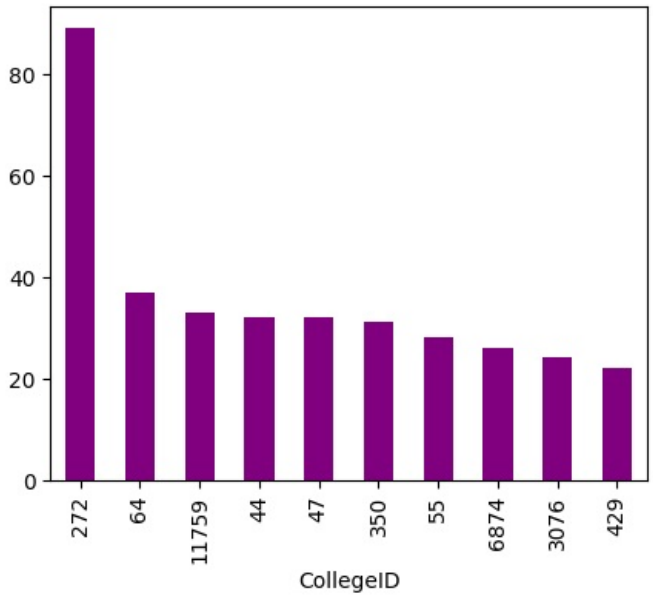
`insight` :

The analysis reveals a notable disparity in the distribution of employees based on college city tier. Specifically, there is a higher representation of employees from college city tier 0 compared to tier 1. Within tier 0, approximately 70.02% of employees originate from this tier, indicating a predominant presence. In contrast, the remaining employees, approximately 29.98%, hail from tier 1 college cities. This observation underscores the uneven distribution of employees across different tiers of college cities, with a larger proportion originating from tier 0 cities. Understanding such disparities in geographical representation is crucial for workforce planning, resource allocation, and addressing potential challenges related to talent acquisition and retention in different geographic regions.

1.6 CollegeID

In [45]:
```
(df1['CollegeID'].sort_values(ascending=False).value_counts().head(10)).plot(kind='bar',color='purple',figsize=
```

Out[45]:
```
<Axes: xlabel='CollegeID'>
```



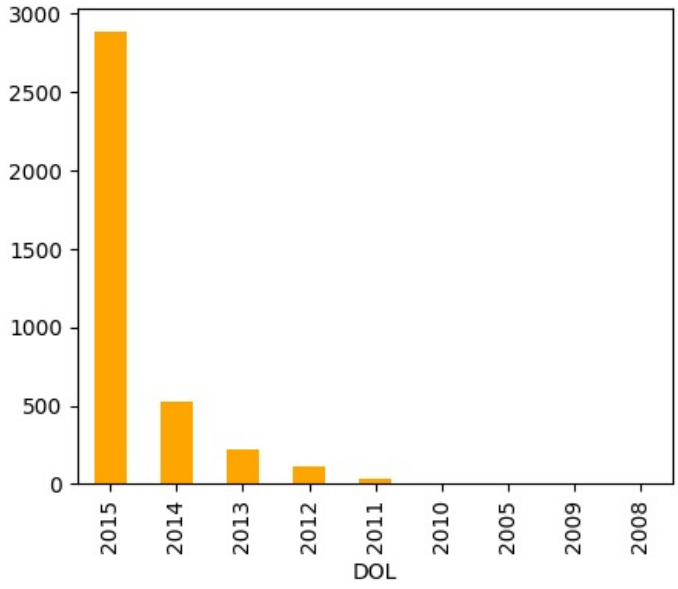
`insight` :

The analysis highlights a significant concentration of individuals originating from College ID 272, with this college having the highest representation among the dataset. Following closely behind, College ID 64 emerges as the second-highest contributor to the dataset, albeit with a lower number of individuals compared to College ID 272. Interestingly, the remaining colleges exhibit similar levels of representation, with relatively consistent numbers of individuals from each college. This observation suggests a pronounced disparity in the representation of candidates across different colleges, with a few colleges contributing a larger number of individuals compared to others. Understanding such variations in college representation can provide valuable insights into recruitment patterns, educational partnerships, and talent pipelines for organizations.

1.6 Date of leaving

In [46]:
```
(df1['DOL'].dt.year.value_counts().head(10)).plot(kind='bar',color='orange',figsize=(5, 4))
```

Out[46]:
```
<Axes: xlabel='DOL'>
```



`insight` :

The analysis reveals notable variations in the rate of job departures across different years within the dataset. Particularly, there is a discernible increase in the number of individuals leaving their jobs in the year 2024, with this year experiencing a relatively high departure rate. Similarly, a comparable trend was observed in the year 2015, indicating another period of heightened job turnover.

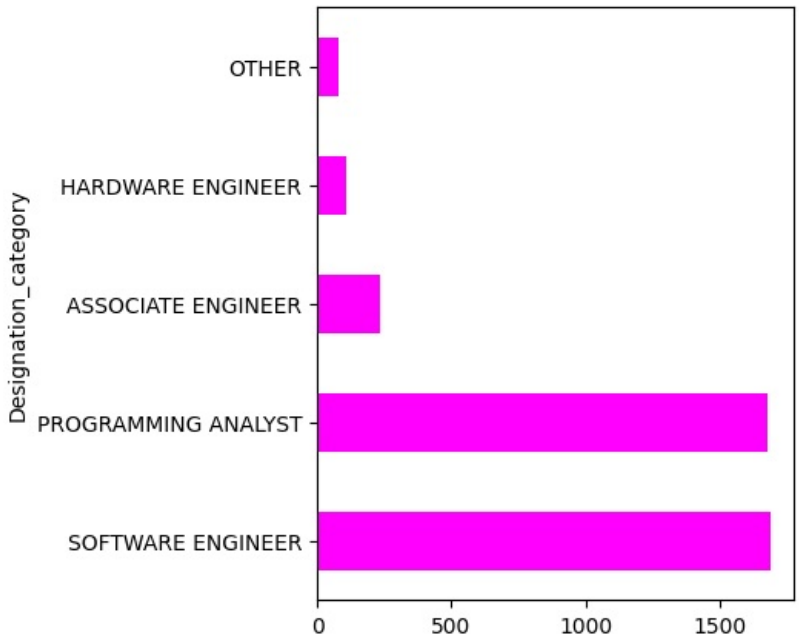
Conversely, the years 2008, 2005, 2008, and 2010 exhibit either minimal or no instances of job departures, suggesting periods of stability or lower turnover within the workforce during those years.

These observations highlight fluctuations in job retention and turnover rates over time, with certain years experiencing higher levels of employee turnover compared to others. Understanding such trends is crucial for organizations to identify potential patterns, address retention challenges, and implement strategies to mitigate turnover and maintain a stable workforce.

1.7 Designation category

In [47]:
```
(df1['Designation_category'].str.upper().value_counts().head(10)).plot(kind='barh',color='magenta',figsize=(4,
```

Out[47]:
```
<Axes: ylabel='Designation_category'>
```



`insight`:

Upon thorough categorization of designations, it has been observed that the employee count is significantly higher in the categories of Programming Analyst and Software Engineer. Conversely, there is a notably smaller representation of individuals in the Hardware Engineer and other categories.

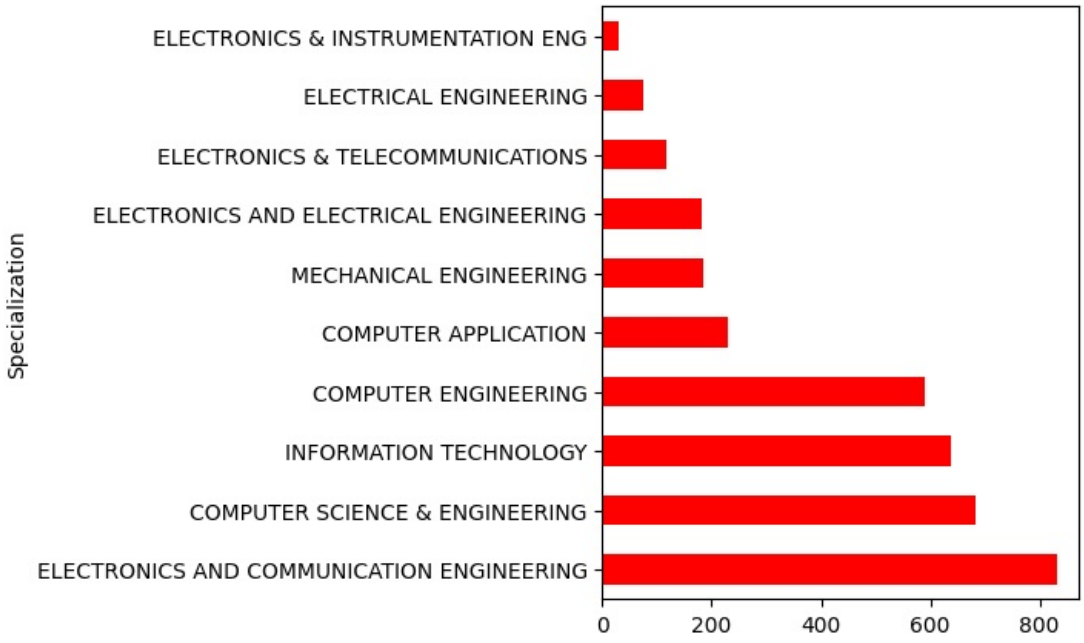
This observation suggests a predominant focus on roles related to programming and software engineering within the dataset, with a larger proportion of employees occupying positions in these domains. In contrast, roles such as Hardware Engineer and other specialized categories have a relatively smaller representation, indicating a lesser prevalence or demand for such roles among the dataset participants.

Understanding these trends in job distribution across different designation categories is valuable for organizations in workforce planning, talent acquisition, and skill development initiatives. It enables organizations to align their hiring strategies with the prevailing job market trends and demand for specific skill sets within the industry.

1.8 Specialization

In [48]:
```
(df1['Specialization'].str.upper().value_counts().head(10)).plot(kind='barh',color='red',figsize=(4, 5))
```

Out[48]:
```
<Axes: ylabel='Specialization'>
```



`insights`:

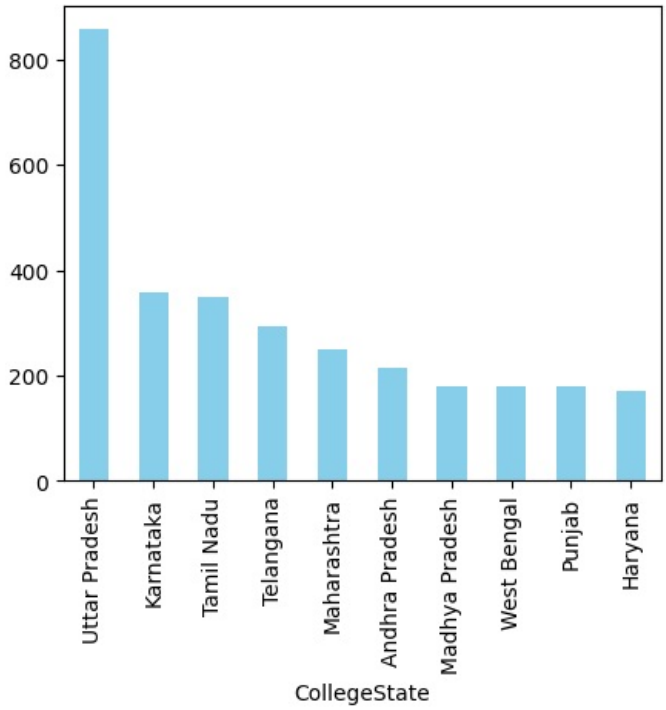
The analysis reveals that the highest number of individuals in the dataset have a background in Electronics and Communication Engineering, followed closely by Computer Science and Engineering. Additionally, there is a notable representation of individuals with backgrounds in Computer Engineering and Information Technology, with these categories having a similar level of representation.

This observation underscores the prevalence of candidates with engineering backgrounds, particularly in disciplines related to electronics, communication, and computer science. These fields are often associated with technology-driven industries and are in high demand in the job market. Understanding the distribution of educational backgrounds among candidates provides insights into the skill sets and expertise available within the talent pool, informing recruitment strategies and skill development initiatives within organizations and industries.

1.9 College State

In [49]:
```
(df1['CollegeState'].value_counts().head(10)).plot(kind='bar',color='skyblue',figsize=(5, 4))
```

Out[49]:
```
<Axes: xlabel='CollegeState'>
```



`insights`:

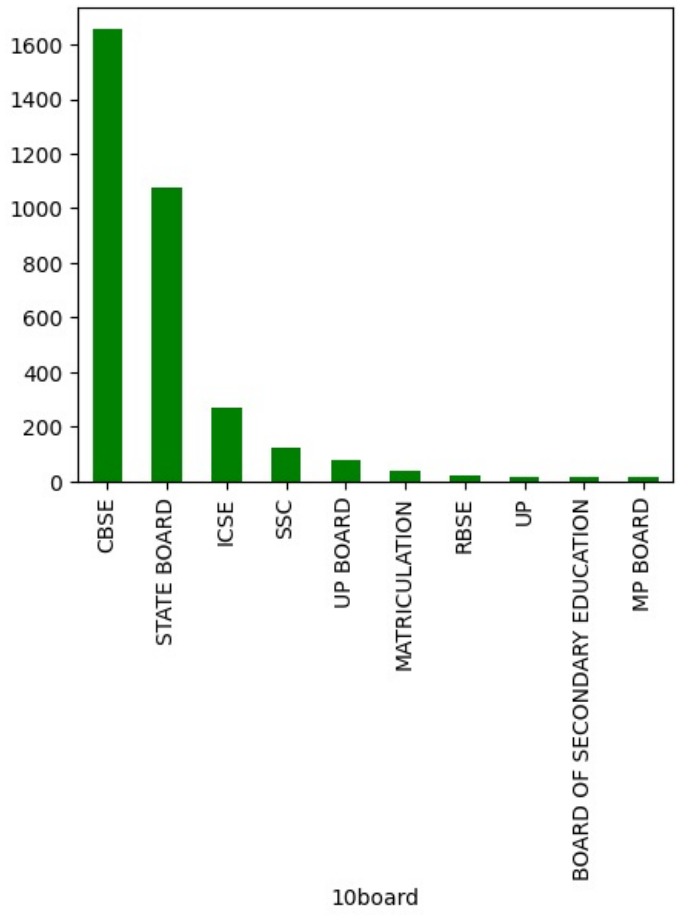
The analysis reveals that the highest number of individuals in the dataset hail from Uttar Pradesh (UP), followed by Karnataka and Tamil Nadu. Notably, Uttar Pradesh exhibits a significantly higher representation compared to Karnataka and Tamil Nadu, despite the latter two being the second and third highest, respectively, in terms of representation.

This observation highlights a substantial concentration of candidates from Uttar Pradesh, suggesting a strong presence of candidates from this region within the dataset. In contrast, while Karnataka and Tamil Nadu are the second and third highest, their representation is relatively lower compared to Uttar Pradesh. Among the remaining states, there is a relatively consistent and lower representation, with no single state exhibiting a significantly higher or lower presence compared to others. Understanding these geographic distribution patterns can provide insights into regional talent pools, recruitment dynamics, and potential areas for targeted talent acquisition strategies. Additionally, it may reflect underlying demographic, economic, or educational factors influencing employment patterns across different states

1.10 10th board

In [50]:
```
(df1['10board'].str.upper().value_counts().head(10)).plot(kind='bar',color='green',figsize=(5, 4))
```

Out[50]:
```
<Axes: xlabel='10board'>
```

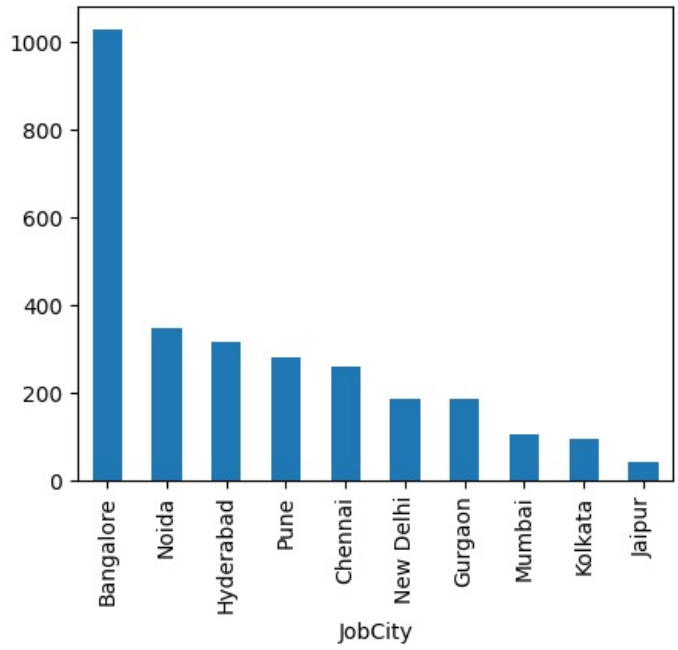


`insight`:

The analysis indicates a similar trend to that of the 12th board, with a higher number of employees coming from the Central Board of Secondary Education (CBSE) background, followed by candidates from their respective state boards as the second highest group. This consistency suggests a preference for CBSE-affiliated schools among employees, with state boards also being a significant contributor to the talent pool. Understanding this pattern can offer insights into the educational backgrounds of employees and potentially inform recruitment strategies or training programs tailored to these specific educational backgrounds.

1.12 Job City

In [51]:
```
(df1['JobCity'].value_counts().head(10)).plot(kind='bar', figsize=(5, 4))
plt.show()
```



`insights` :

The analysis reveals that Bangalore has the highest number of individuals in the dataset, followed by Noida and Hyderabad as the second and third highest, respectively. Notably, while Noida and Hyderabad rank second, their representation is not as high as Bangalore. Additionally, the remaining states show relatively similar levels of representation, without any single state significantly surpassing the others. This observation underscores Bangalore's prominence as a major hub for employment opportunities, with Noida and Hyderabad also emerging as significant contributors, albeit to a lesser extent. Understanding these regional distribution patterns can provide insights into geographical preferences among employees and guide strategic decisions related to talent acquisition and resource allocation in different locations.

Bivariate Analysis

2.1 Salary and Gender

In [52]:
```
# Numeric vs. Categorical (Violin plot)
sns.violinplot(y='Gender', x='Salary', data=df1,color='yellow')
plt.title('Salary by Gender')


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

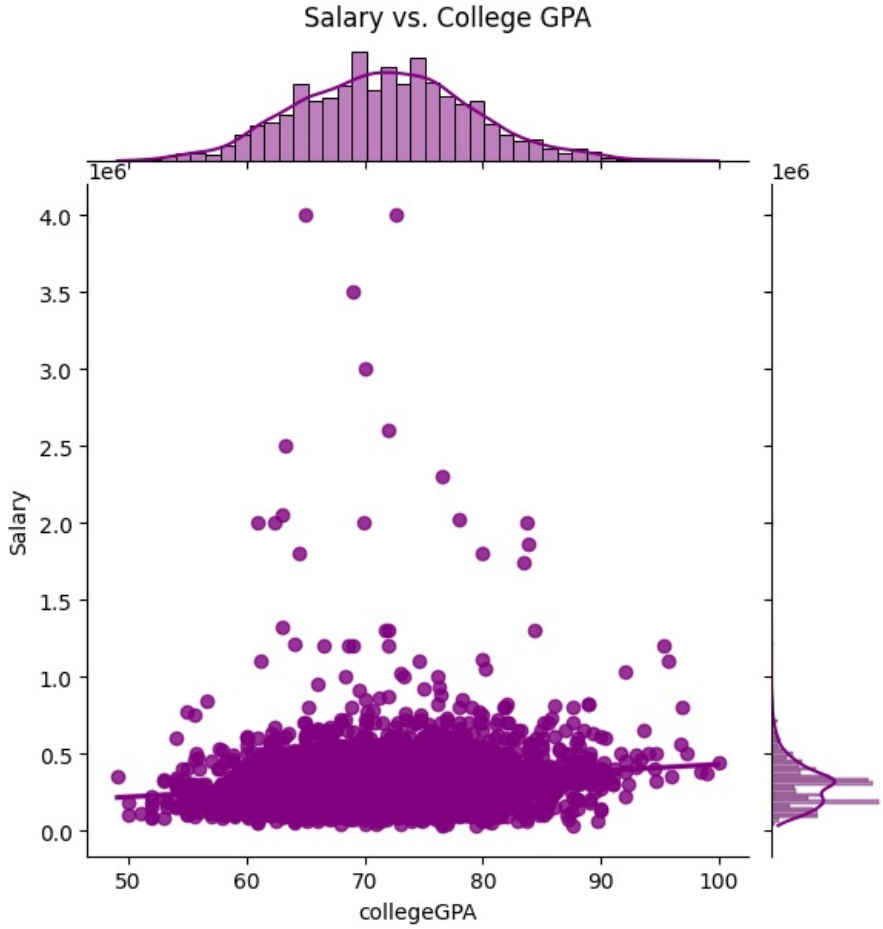
`insight` :

In the violin plot analysis, it's evident that there's minimal disparity in salaries between males and females. Both genders exhibit similar median salary lines, indicating comparable central tendencies in their salary distributions. Additionally, the shape and spread of the data points appear consistent between males and females, suggesting a similar distribution pattern across gender groups. Overall, these findings suggest that, within this dataset, gender does not appear to be a significant factor influencing salary differentials.

2.2 salary and CollegeGpa

In [53]:
```
# Numeric vs. Numeric (Joint plot)
sns.jointplot(y='Salary', x='collegeGPA', data=df1, kind='reg',color='purple')
plt.suptitle('Salary vs. College GPA', y=1.02)
```

Out[53]:
```
Text(0.5, 1.02, 'Salary vs. College GPA')
```



`insight`:

The majority of data points are clustered at the lower end of the salary scale, indicating that many individuals with varying GPAs have similar salaries. There is no clear trend or correlation visible, suggesting that college GPA may not be a strong predictor of salary. The histograms on the top and right sides of the scatter plot show the distribution of college GPA and salary respectively. Most data points are concentrated at the lower end of the salary scale, forming a dense cluster. A few scattered data points represent individuals with higher salaries, but there's no clear pattern relating these to college GPA. This suggests that while GPA is an important factor in academic performance, it may not directly correlate with salary, and other factors could be at play.

2.3 Salary and 10th percentage

In [54]:
```
# Numeric vs. Numeric (Scatter plot)
sns.scatterplot(y='Salary', x='10percentage', data=df1,color='green')
plt.title('Salary vs. 10th Percentage')
plt.figure(figsize=(5, 4))
plt.show()
```

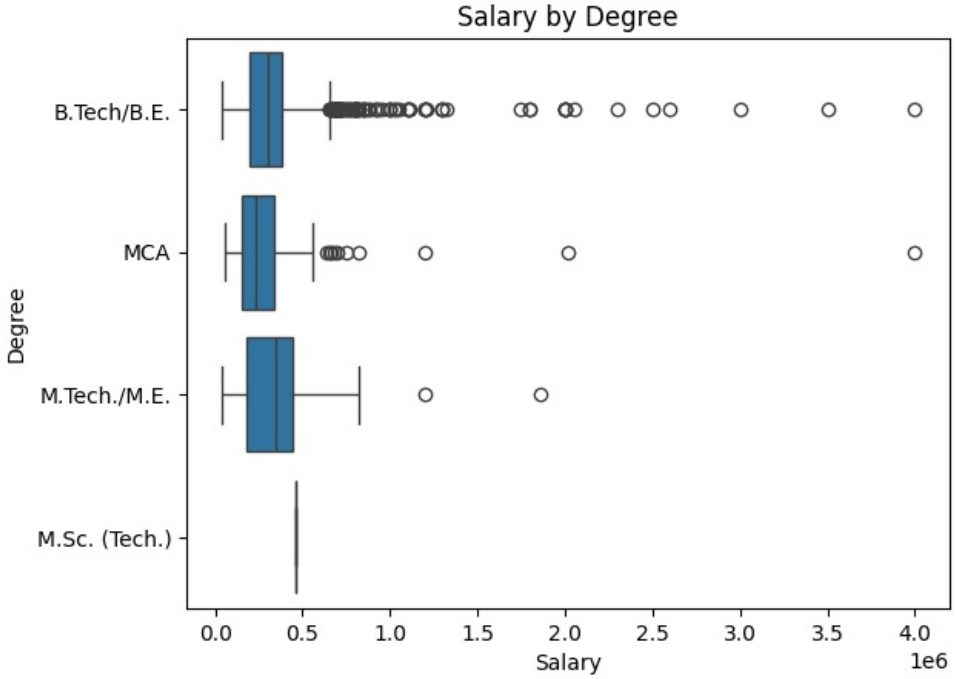

```
<Figure size 500x400 with 0 Axes>
```

`insight` :

The scatter plot reveals a predominant concentration of individuals with higher percentages compared to those with lower percentages. Notably, salary levels appear relatively consistent across different percentage ranges, indicating a uniform distribution of salaries irrespective of academic performance. However, intriguingly, there are outliers where individuals with lower percentages exhibit higher salaries. This observation suggests the presence of additional influential factors beyond academic achievements in determining salary outcomes, underscoring the multifaceted nature of salary determinants.

2.4 Salary and Degree

In [55]:
```
# Numeric vs. Categorical (Box plot)
sns.boxplot(y='Degree', x='Salary', data=df1)
plt.title('Salary by Degree')
plt.figure(figsize=(5, 4))
plt.show()
```




```
<Figure size 500x400 with 0 Axes>
```

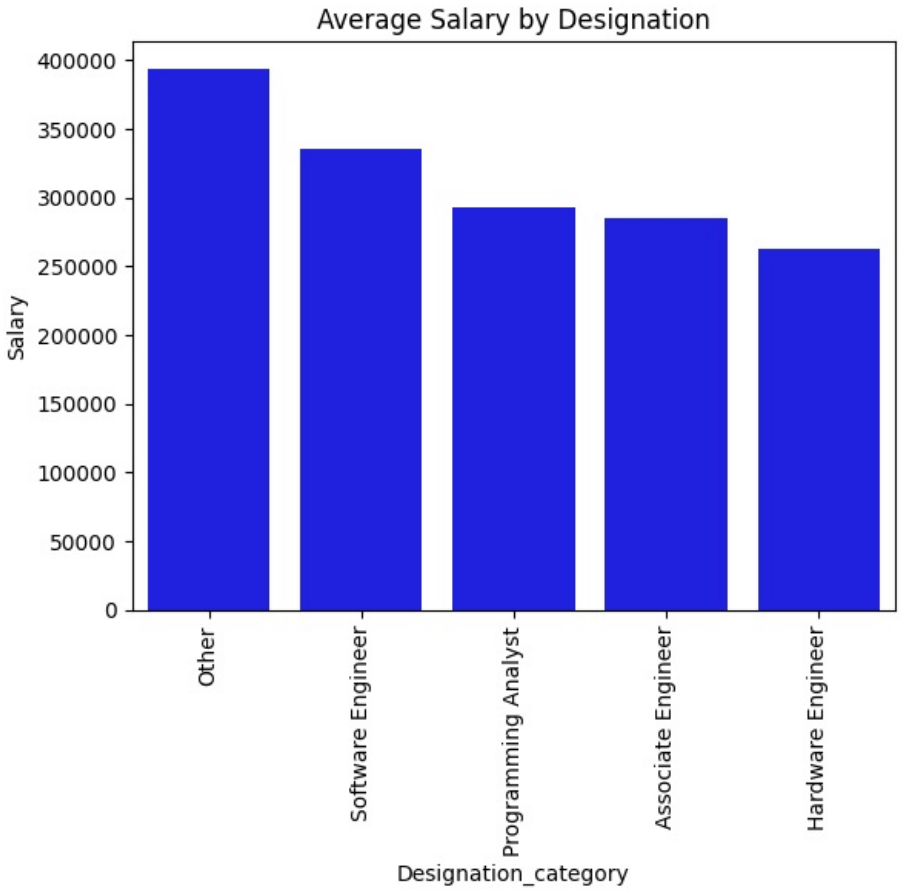
`insight` :

The plot demonstrates several outliers among individuals with a BTech/BE background, indicating instances where salaries are notably high despite this educational qualification. Additionally, upon closer examination, individuals with an MTech/ME background exhibit a higher median salary compared to other educational backgrounds. Furthermore, individuals with an MCA degree also display some outliers, suggesting that they too can achieve relatively high salaries. Overall, these insights underscore the diverse salary distributions

across different educational backgrounds, with certain outliers highlighting the potential for exceptional earning opportunities within specific qualifications.

2.5 Salary and Designation

In [56]:
```
# Categorical vs. Categorical (Stacked bar plot)
salary_by_desn = df1.groupby('Designation_category')['Salary'].mean().reset_index().sort_values(by='Salary',asc
sns.barplot(x='Designation_category', y='Salary', data=salary_by_desn, ci=None,color='blue')
plt.title('Average Salary by Designation')

plt.xticks(rotation=90)
plt.show()
```

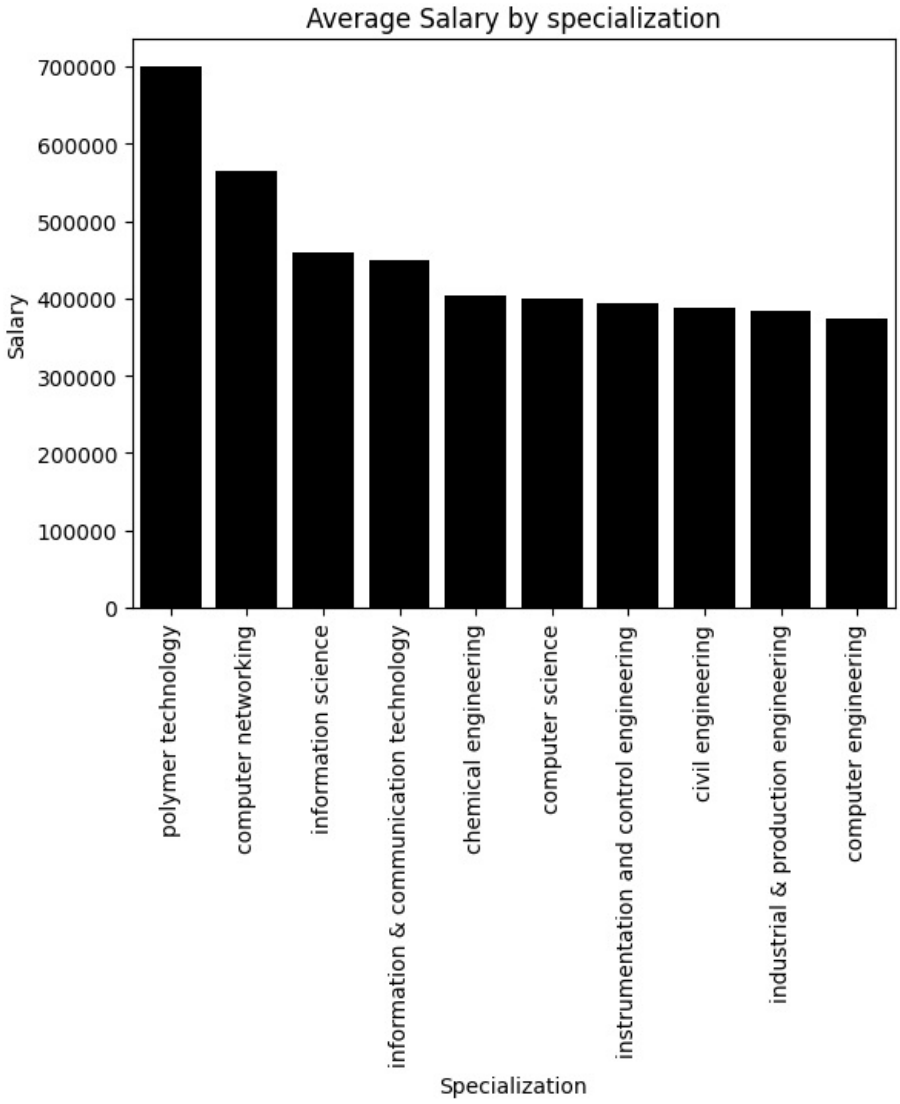


`insight` :

The analysis reveals that individuals falling into the "Other" category exhibit a higher average salary compared to other designation categories. Furthermore, among the top four categories, software engineers command the highest average salaries, followed by programming analysts. This observation suggests that software engineering roles tend to offer more lucrative salary prospects compared to other job titles within the dataset.

2.6 salary and specialization

In [57]:
```
# Categorical vs. Categorical (Stacked bar plot)
salary_by_sp = df1.groupby('Specialization')['Salary'].mean().reset_index().sort_values(by='Salary',ascending=F
sns.barplot(x='Specialization', y='Salary', data=salary_by_sp, ci=None,color='black')
plt.title('Average Salary by specialization')

plt.xticks(rotation=90)
plt.show()
```



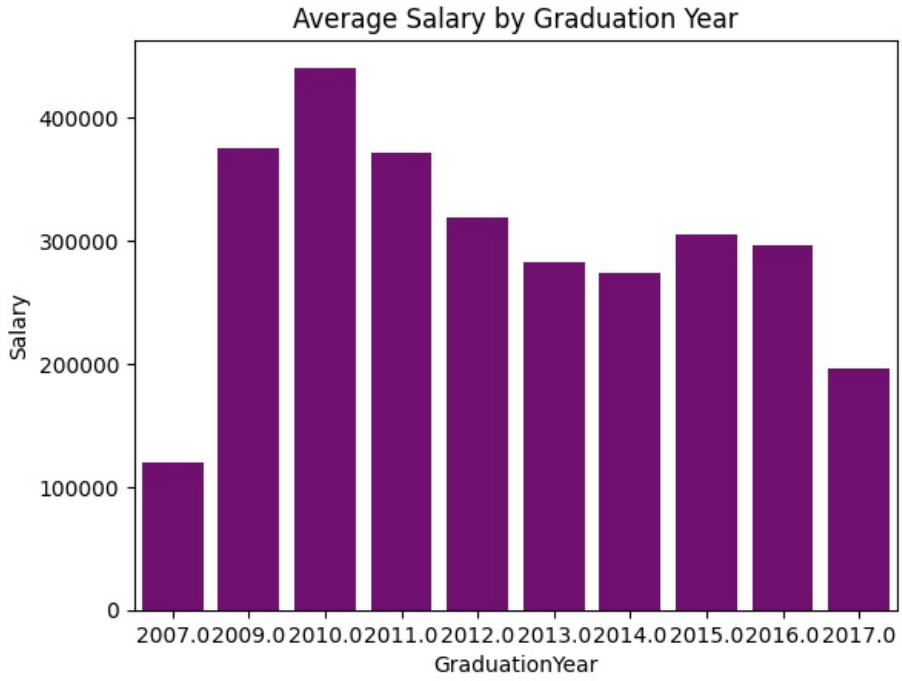
`insight`:

The analysis reveals that individuals specializing in polymer technology tend to have the highest average salary, followed closely by those specializing in computer networking. Notably, when considering the broader spectrum of specializations, there appears to be a consistent trend of favorable average salaries across all areas. This suggests that professionals with various specializations have the potential to secure competitive salaries within their respective fields of expertise, underscoring the overall positive outlook for earnings among individuals with diverse skill sets and areas of focus.

2.7 salary and Graduation Year

In [74]:
```
# Categorical vs. Categorical (Stacked bar plot)
salary_by_gy = df1.groupby('GraduationYear')['Salary'].mean().reset_index().sort_values(by='Salary',ascending=F
sns.barplot(x='GraduationYear', y='Salary', data=salary_by_gy, ci=None,color='purple')
plt.title('Average Salary by Graduation Year')
```

Out[74]:
```
Text(0.5, 1.0, 'Average Salary by Graduation Year')
```



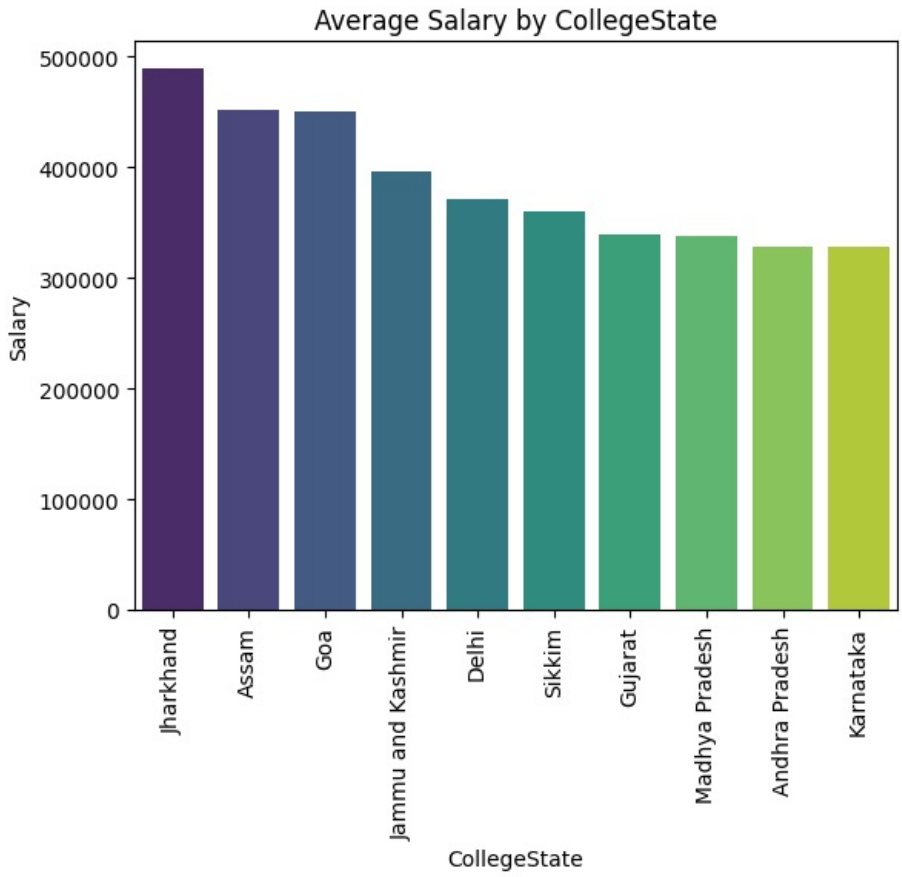
`insight`:

The trend in average salaries exhibits a notable increase from 2007 to 2010, indicating a period of growth and potentially favorable economic conditions. However, this upward trajectory is followed by a sharp decline until 2014, suggesting a period of economic downturn or other factors impacting salary levels. Interestingly, there is a slight uptick in average salaries observed in 2016, albeit not sufficient to fully offset the preceding decline.

Overall, these fluctuations in average salaries reflect a degree of inconsistency in salary trends over the years under analysis. This variability may be influenced by various factors such as changes in market demand, economic cycles, industry trends, and shifts in employer preferences or hiring practices. Further analysis and consideration of contextual factors are necessary to better understand the underlying reasons behind these fluctuations and their implications for the job market.

2.8 salary And College State

In [73]:
```
# Categorical vs. Categorical (Stacked bar plot)
salary_by_designation = df1.groupby('CollegeState')['Salary'].mean().reset_index().sort_values(by='Salary',asce
sns.barplot(x='CollegeState', y='Salary', data=salary_by_designation, ci=None,palette='viridis')
plt.title('Average Salary by CollegeState')

plt.xticks(rotation=90)
plt.show()
```



`insights`:

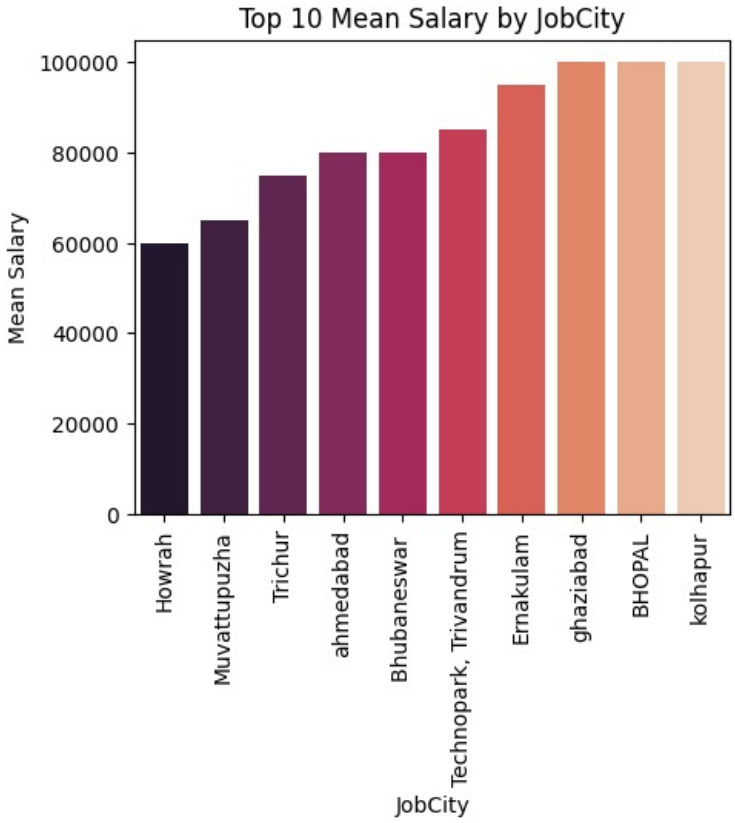
The analysis indicates that students who study in colleges located in Jharkhand, Assam, and Goa tend to have relatively higher average salaries compared to students from colleges in other states. This observation suggests a correlation between the geographic location of colleges and the average salaries of their graduates.

Jharkhand, Assam, and Goa stand out as states where students from colleges within these regions achieve higher levels of salary upon graduation. This trend may be influenced by several factors, including regional economic conditions, industry presence, job market dynamics, and the quality of education offered by colleges in these states.

The higher average salaries observed in colleges located in Jharkhand, Assam, and Goa may attract students seeking better employment opportunities and career prospects. Additionally, it underscores the importance of considering regional factors and local job markets when evaluating college choices and career pathways.

2.9 salary and Job City

In [70]:
```
salary_by_jc = df.groupby('JobCity')['Salary'].mean().reset_index()

# Sort the data in ascending order by mean salary
salary_by_jc = salary_by_jc.sort_values(by='Salary', ascending=True)

# Select the top 10 values
top_10_jc = salary_by_jc.head(10)

# Extract the top 10 board names
top_10_boards = top_10_jc['JobCity']

# Plot the top 10 values
plt.figure(figsize=(5, 4))
sns.barplot(x='JobCity', y='Salary', data=top_10_jc, order=top_10_boards,palette='rocket')
plt.title('Top 10 Mean Salary by JobCity')
plt.xlabel('JobCity')
plt.ylabel('Mean Salary')
plt.xticks(rotation=90)
plt.show()
```

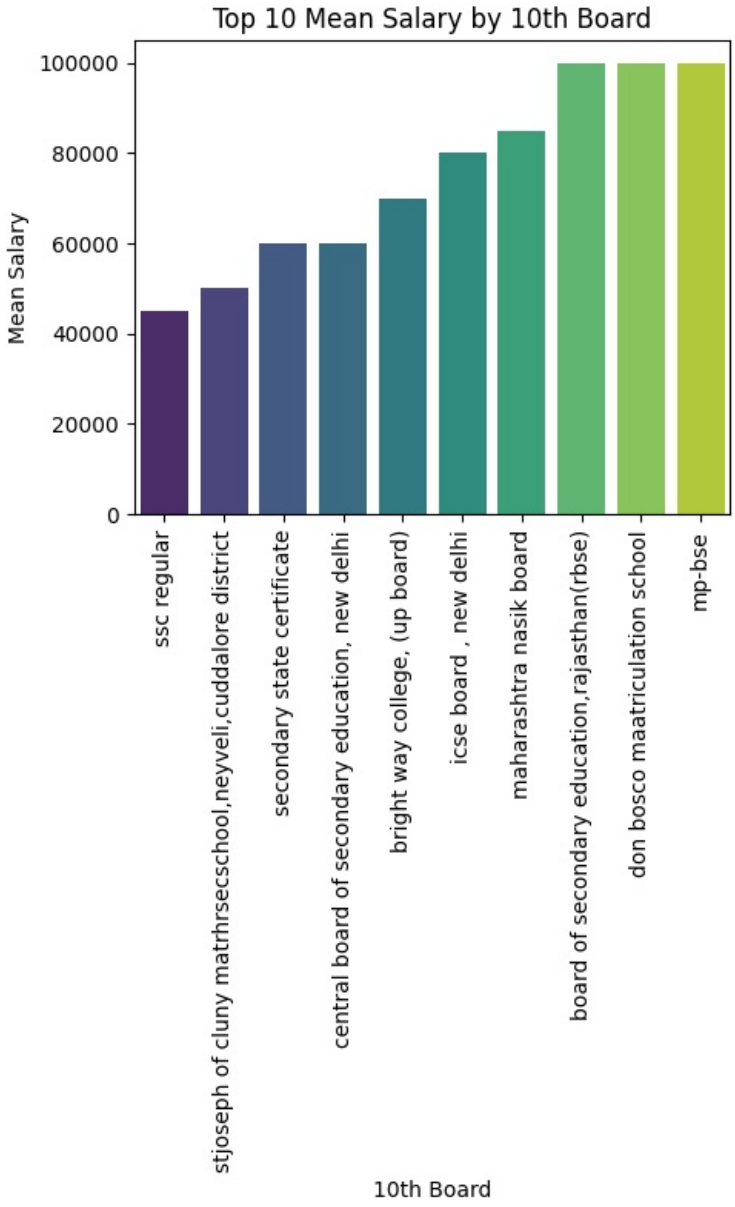


`insight`:

The analysis reveals that job cities like Kolhapur, Bhopal, and Ghaziabad boast higher average salaries, with Kochi trailing closely behind. These cities serve as thriving employment hubs where professionals command competitive compensation. Their robust job markets and attractive opportunities make them sought-after destinations for skilled workers. Understanding the factors behind their economic success can offer valuable insights for stakeholders and policymakers alike.

2.10 salary and 10th board

In [69]:
```
salary_by_10board = df1.groupby('10board')['Salary'].mean().reset_index()

# Sort the data in ascending order by mean salary
salary_by_10board = salary_by_10board.sort_values(by='Salary', ascending=True)

# Select the top 10 values
top_10_10board = salary_by_10board.head(10)

# Extract the top 10 board names
top_10_boards = top_10_10board['10board']

# Plot the top 10 values
plt.figure(figsize=(5, 4))
sns.barplot(x='10board', y='Salary', data=top_10_10board, order=top_10_boards,palette='viridis')
plt.title('Top 10 Mean Salary by 10th Board')
plt.xlabel('10th Board')
plt.ylabel('Mean Salary')
plt.xticks(rotation=90)
plt.show()
```



`insight` :

These three institutions, St. Mary Higher Secondary, Board of Secondary Education, Rajasthan (RBSE), and Karnataka State Secondary Education Board (KSSEB), stand out for offering notably high average salaries compared to other 10th boards. This observation underscores the potential influence of educational backgrounds on salary outcomes, highlighting the significance of academic credentials in shaping employment opportunities and compensation levels.

In [62]:
```
top_10_10board
```

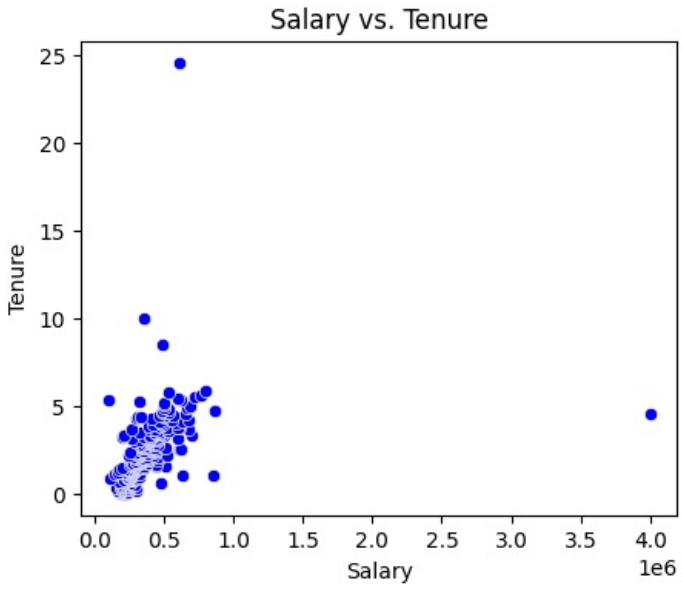
Out[62]:

	10board	Salary
215	ssc regular	45000.0
236	stjoseph of cluny matrhrsecschool,neyveli,cudd...	50000.0
210	secondary state certificate	60000.0
64	central board of secondary education, new delhi	60000.0
44	bright way college, (up board)	70000.0
96	icse board , new delhi	80000.0
151	maharashtra nasik board	85000.0
38	board of secondary education,rajasthan(rbse)	100000.0
75	don bosco maatriculation school	100000.0
177	mp-bse	100000.0

2.11 salary and Tenure

In [63]:
```
# Group by Tenure and calculate the mean salary
salary_by_Tenure = df1.groupby('Tenure')['Salary'].mean().reset_index()

# Set figure size
```

```
plt.figure(figsize=(5, 4))  # Adjust the figure size as needed

# Scatter plot of Salary vs. Tenure
sns.scatterplot(x='Salary', y='Tenure', data=salary_by_Tenure, color='blue')
plt.title('Salary vs. Tenure')
plt.xlabel('Salary')
plt.ylabel('Tenure')
plt.show()
```

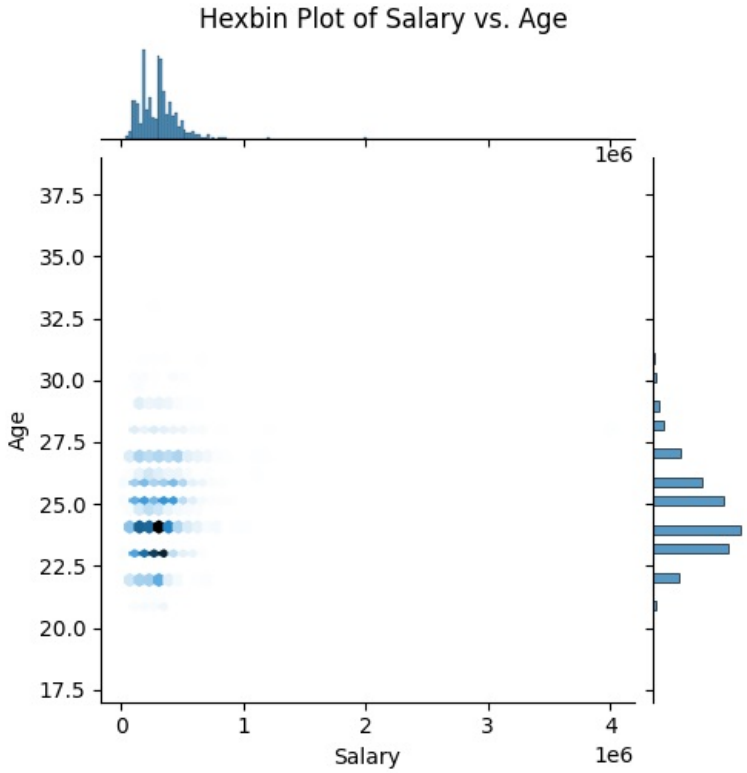


insight:

The plot shows a positive correlation between salary and tenure, meaning as tenure increases, the salary tends to increase as well. However, there is a wide spread in the data points indicating variability in salaries at similar levels of tenure. A concentration of data points is visible at the lower left corner, indicating many individuals with low tenure and low salary. As we move right (increasing salary) and up (increasing tenure) on the graph, the number of data points decreases but shows an upward trend. This suggests that while tenure is an important factor in salary, there is a significant variability in salaries at similar levels of tenure

2.12 Salary and Age

In [64]:
```
sns.jointplot( x='Salary', y='Age',data=df1,kind='hex', height=5)
plt.suptitle('Hexbin Plot of Salary vs. Age', y=1.02)

plt.show()
```

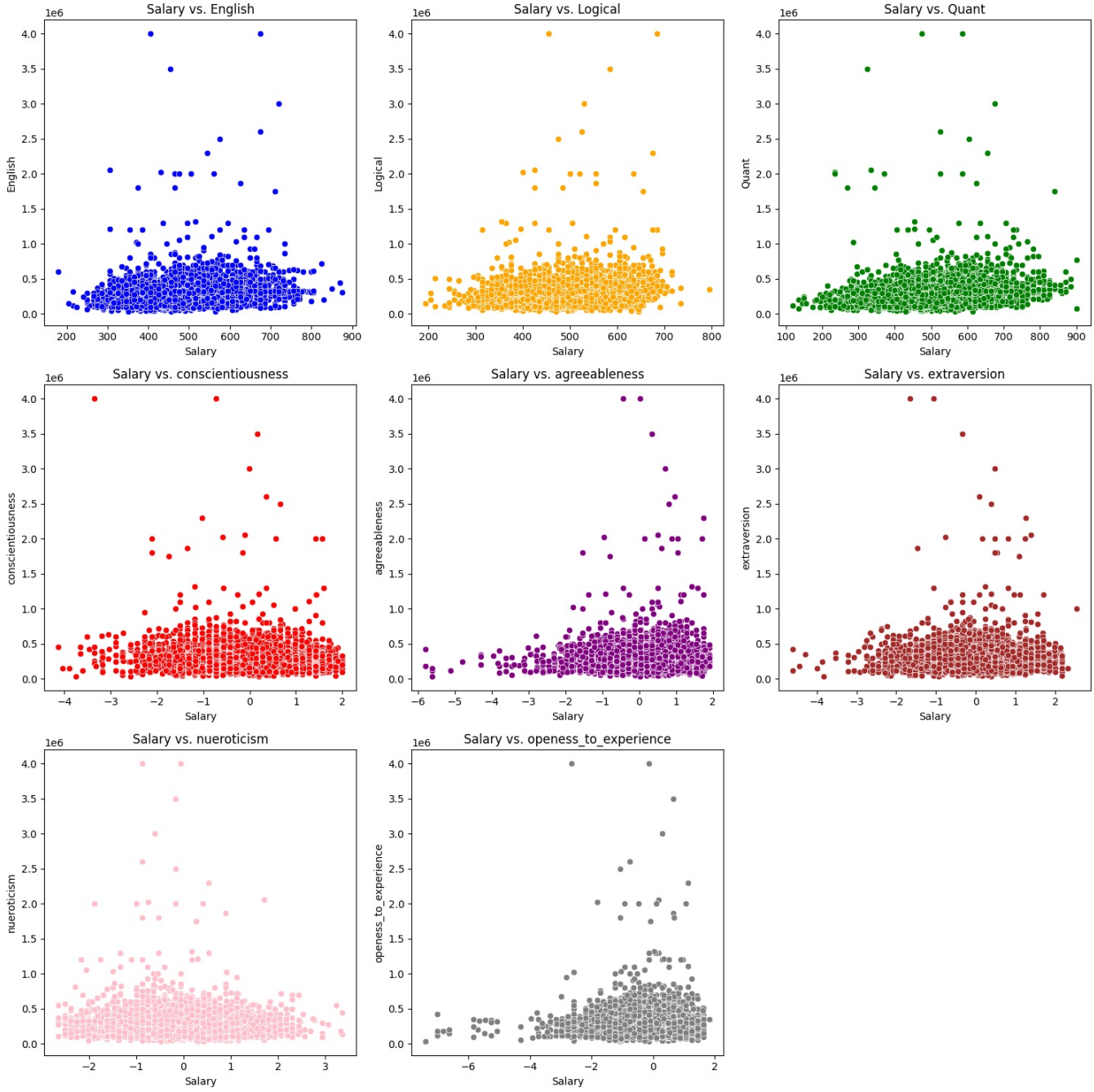


insight:

The plot shows a distribution of salaries versus age. The majority of the data points are concentrated around the lower salary range for individuals aged between 22.5 and 27.5 years old. There are blue hexagonal bins representing concentrations of data points; darker bins indicate higher concentrations. A histogram at the top shows salary distribution, with most people earning less. A histogram on the right side shows age distribution, indicating more younger individuals in this dataset. This suggests that while age is an important factor in salary, there is a significant variability in salaries at similar ages.

2.13 Salary and'English', 'Logical', 'Quant', 'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience'

In [65]:
```
# Define the columns to plot against 'Salary'
columns_to_plot = ['English', 'Logical', 'Quant', 'conscientiousness', 'agreeableness', 'extraversion', 'nuerot

# Define a list of colors to use for each plot
colors = ['blue', 'orange', 'green', 'red', 'purple', 'brown', 'pink', 'gray']

# Calculate the number of rows and columns for the grid
num_plots = len(columns_to_plot)
num_cols = 3
num_rows = (num_plots + num_cols - 1) // num_cols

# Create a grid of subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 15))

# Flatten the axes array for easier iteration
axes = axes.flatten()

# Iterate over each column and create a scatter plot with a specified color
for i, (column, color) in enumerate(zip(columns_to_plot, colors)):
    sns.scatterplot(y='Salary', x=column, data=df1, ax=axes[i], color=color)
    axes[i].set_title(f'Salary vs. {column}')
    axes[i].set_xlabel('Salary')
    axes[i].set_ylabel(column)

# Hide empty subplots
for i in range(num_plots, num_rows * num_cols):
    fig.delaxes(axes[i])

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



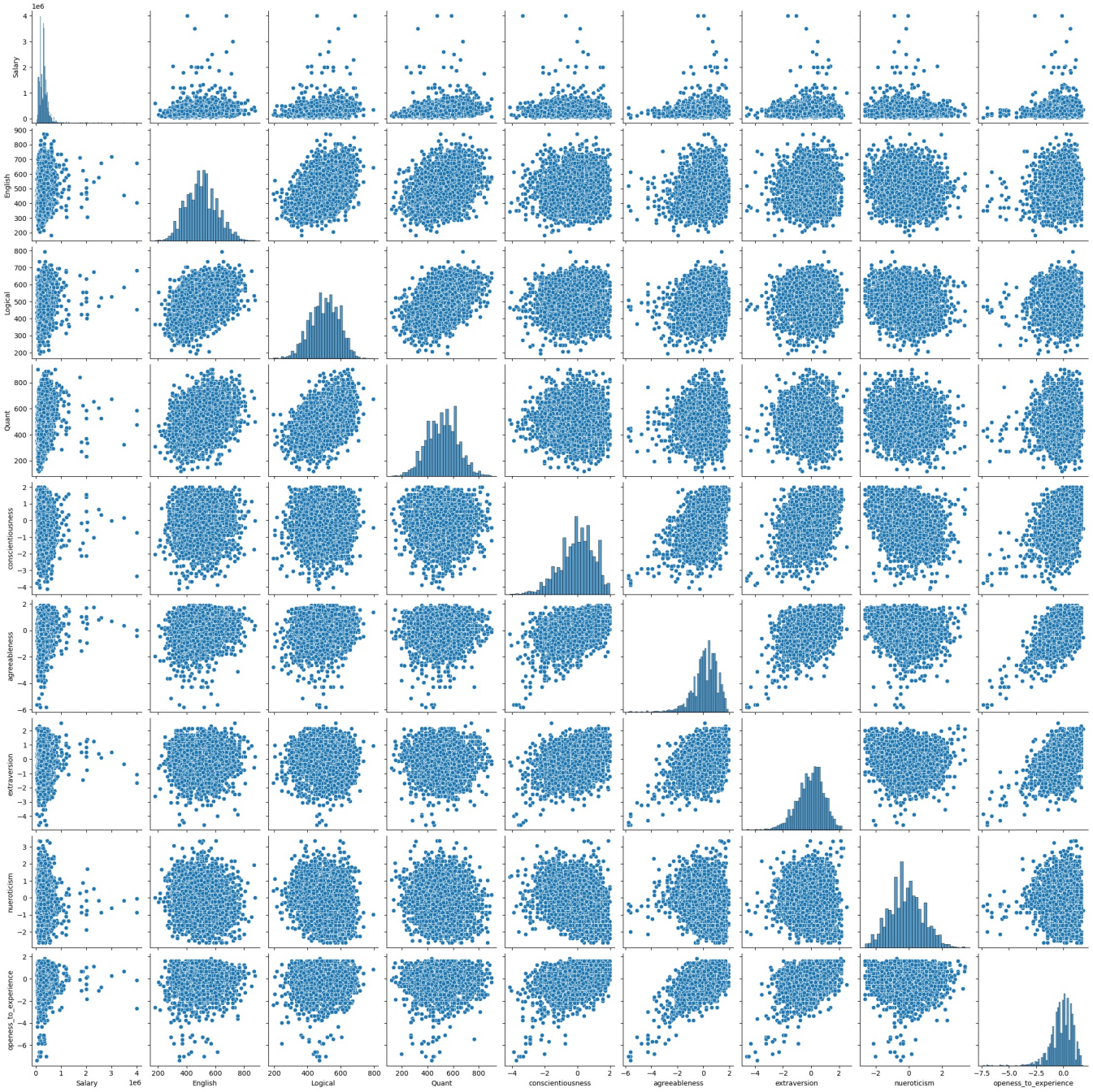
`insight`:

The scatter plots for the columns 'English', 'Logical', 'Quant', 'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism', and 'openess_to_experience' reveal a consistent pattern where the points are spread across all ranges of salaries and trend downwards. This suggests that these scores do not have a significant impact on salary levels, as evidenced by the lack of discernible correlation between these factors and salary outcomes. Despite variations in scores, individuals with different levels of performance in these areas appear to receive similar salary offers. This observation implies that other factors beyond cognitive skills, personality traits, and domain-specific knowledge may play a more influential role in determining salary levels in the dataset.

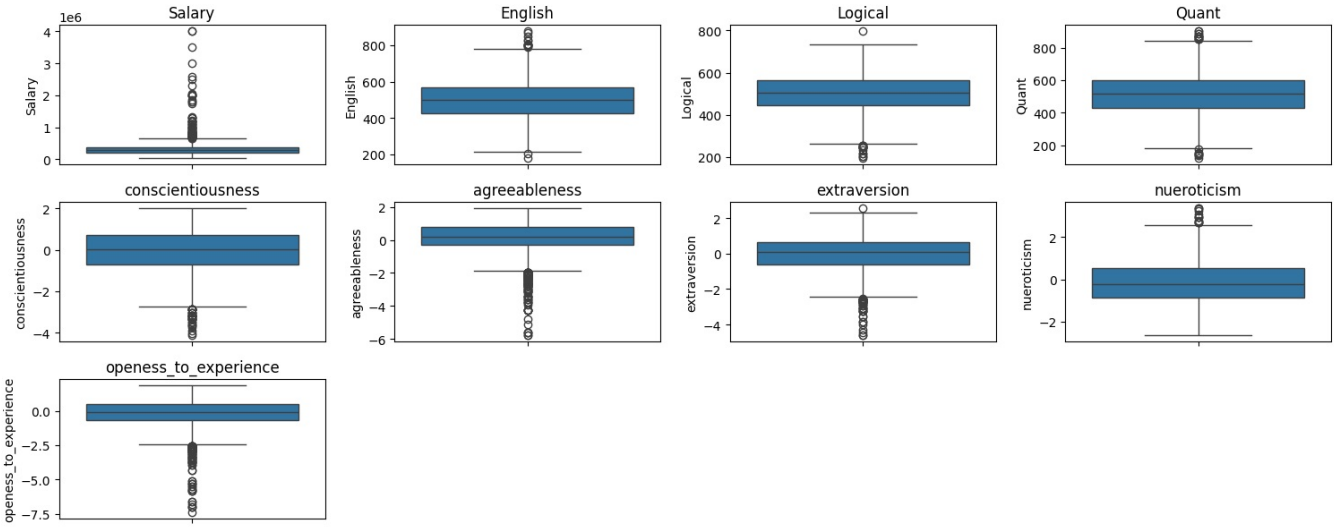
Multi-Variate Analysis

In [78]:
```
top_columns = ['Salary', 'English', 'Logical', 'Quant', 'conscientiousness',
               'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience']

df1_top = df1[top_columns]

# Replace -1 values with NaN
df1_top.replace(-1, np.nan, inplace=True)
plt.figure(figsize=(5, 5))
# Pairplot for numeric columns
sns.pairplot(df1_top.dropna())
plt.show()
```

```
<Figure size 500x500 with 0 Axes>
```

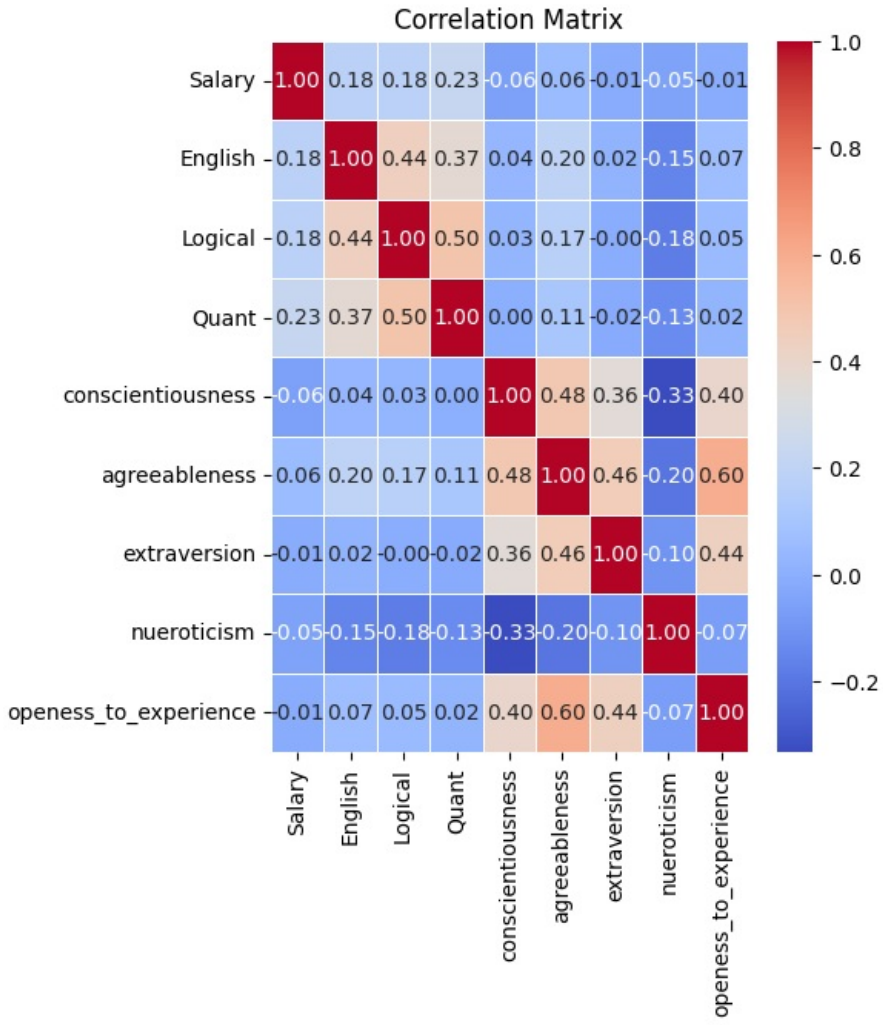


In [79]:
```
# Boxplots for numeric columns
plt.figure(figsize=(15, 10))
for i, col in enumerate(df1_top.columns):
    plt.subplot(5, 4, i+1)
    sns.boxplot(y=df1_top[col].dropna())
    plt.title(col)
plt.tight_layout()
plt.show()
```



In [ ]:

In [80]:
```
# Heatmap for correlation matrix of numeric columns
plt.figure(figsize=(5, 6))
sns.heatmap(df1_top.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix')
plt.show()
```

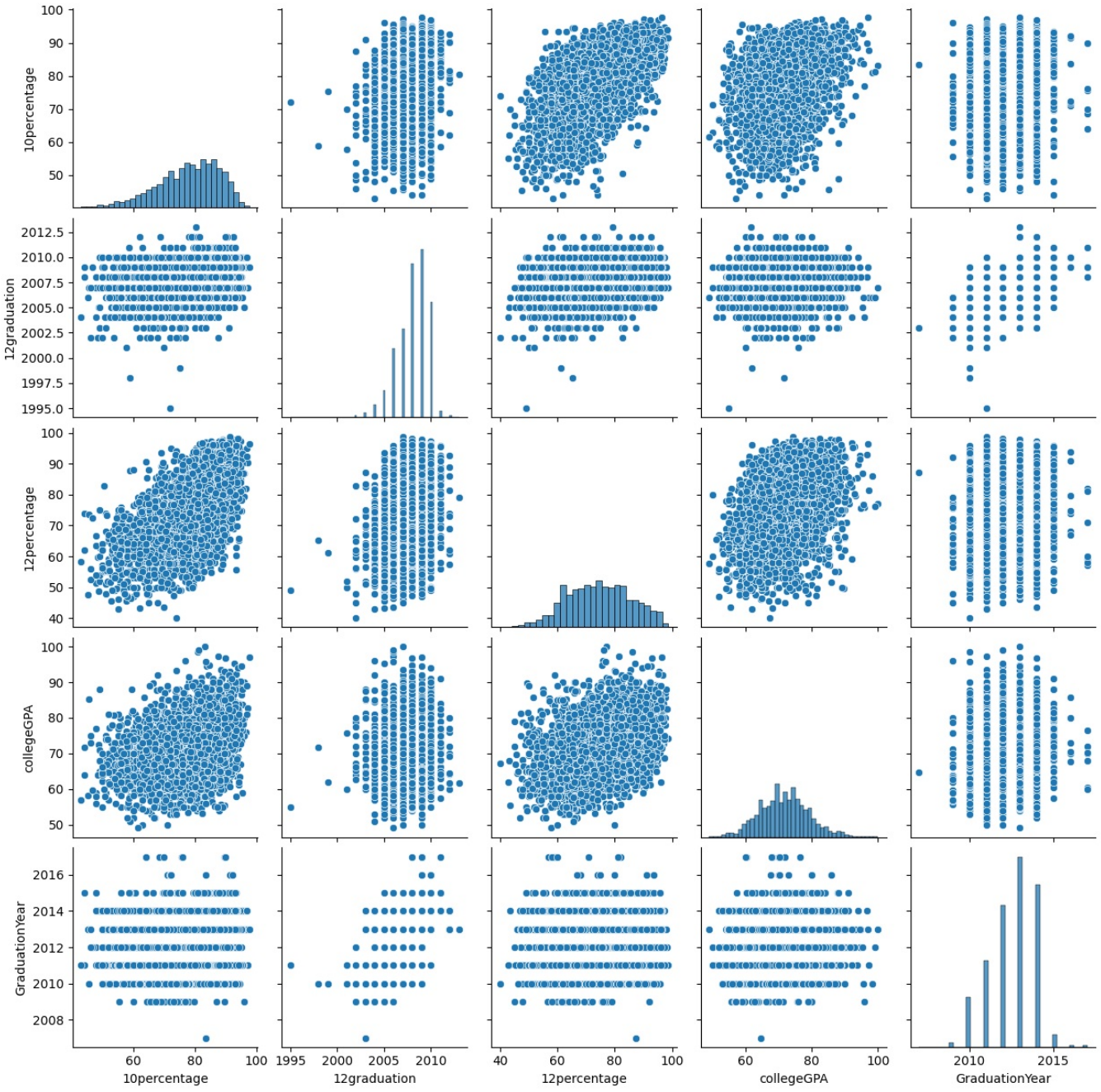


`insights` :

- The matrix displays the correlations between nine different variables: Salary, English proficiency, logical reasoning ability, quantitative skills (Quant), conscientiousness, agreeableness, extraversion, neuroticism, and openness to experience.
- The cells in the matrix are color-coded according to their correlation coefficients; red indicates a strong positive correlation (up to 1.0), blue indicates a strong negative correlation (down to -1.0), and white represents no significant correlation.
- Each variable correlates perfectly with itself as indicated by the red diagonal line from the top left corner to bottom right corner of the matrix where all values are 1.00.
- Salary has a positive correlation with Logical and Quant but is negatively correlated with English.
- English has positive correlations with all personality traits except for Neuroticism.
- Logical and Quant are highly correlated.

- Personality traits like Conscientiousness and Agreeableness have positive correlations among themselves.
- This suggests that while some skills and personality traits are positively correlated with salary, others may not have a significant impact. It's also important to note that correlation does not imply causation, and these relationships could be influenced by other factors not represented in this plot.

In [81]:
```
top_columns = [ '10percentage', '12graduation', '12percentage', 'collegeGPA',
                'GraduationYear']

df_top = df1[top_columns]

# Replace -1 values with NaN
df_top.replace(-1, np.nan, inplace=True)

# Pairplot for numeric columns
sns.pairplot(df_top.dropna())
plt.show()
```



In [ ]:
```

```

solving research question

a)After doing your Computer Science Engineering if you take up jobs as a Programming Analyst, Software Engineer, Hardware Engineer and Associate Engineer you can earn up to 2.5-3 lakhs as a fresh graduate.

In [82]:
```
df3 = df1[(df1["Designation"].isin(["programmer analyst", "software engineer", "hardware engineer", "associate
                (df1["Specialization"].isin(["computer science & engineering", "computer engineering"]))]
```

In [83]:
```
df4=df3[df3['Tenure']<1]
```

In [83]:
```
df4=df3[df3['Tenure']<1]
```

In [84]:
```
(df4['Salary'].mean()/100000).round(2)
```

Out[84]:
```
2.84
```

Null Hypothesis(h1): After doing your Computer Science Engineering if you take up jobs as a Programming Analyst, Software Engineer, Hardware Engineer and Associate Engineer you can earn up to 2.5-3 lakhs as a fresh graduate.¶

Alternative Hypothesis(h0):After doing your Computer Science Engineering if you take up jobs as a Programming Analyst, Software Engineer, Hardware Engineer and Associate Engineer you can't earn up to 2.5-3 lakhs as a fresh graduate.¶

In [85]:
```
from scipy.stats import chi2_contingency

# Define the salary categories
salary_categories = pd.cut(df['Salary'], bins=[0, 250000, 300000, float('inf')], labels=['< 2.5 lakhs', '2.5-3

# Create a contingency table
contingency_table = pd.crosstab(index=df4['Designation'], columns=salary_categories)

# Perform chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Print chi-square test result
print("Chi-square statistic:", chi2)
print("p-value:", p)
```

```
Chi-square statistic: 7.580971659919028
p-value: 0.022584626896985968
```

In [94]:
```
confidence_level = 0.95
alpha = 1 - confidence_level

chi2_critical = chi2.ppf(1 - alpha, dof)

print(chi2_critical)
if(chi2_stat > chi2_critical):
    print("Failed to reject the Null Hypothesis")
else:
    print(" reject the Null Hypothesis")
```

```
61.65623337627955
Failed to reject the Null Hypothesis
```

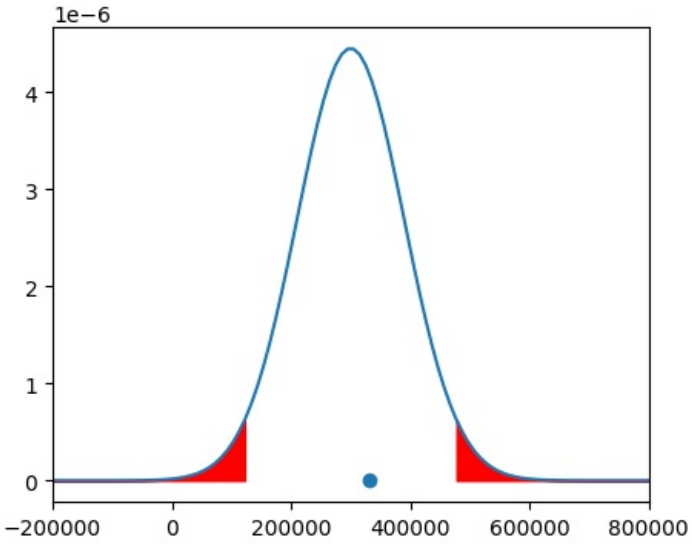
In [99]:
```
pop_mean = 300000
sample_std=89621.3
import statistics
from scipy.stats import t,norm
confidence_level = 0.95

alpha = 1 - confidence_level

t_critical = t.ppf(1 - alpha/2,df=99)

print(t_critical)
```

```
1.9842169515086827
```

In [101…
```
import statistics
from scipy.stats import t,norm
sample_size = 100
sample_mean =332250.0
pop_mean = 300000
sample_std=89621.3

h_min = -200000
h_max = 800000

mean = pop_mean
std = sample_std
plt.figure(figsize=(5,4))
x = np.linspace(h_min, h_max, 100)
y = norm.pdf(x, mean, std)
plt.xlim(h_min, h_max)
plt.plot(x, y)

t_critical_left = pop_mean + (-t_critical * std)
t_critical_right = pop_mean + (t_critical * std)

x1 = np.linspace(h_min, t_critical_left, 100)
y1 = norm.pdf(x1, mean, std)
plt.fill_between(x1, y1, color='red')

x2 = np.linspace(t_critical_right, h_max, 100)
y2 = norm.pdf(x2, mean, std)
plt.fill_between(x2, y2, color='red')
```

```
plt.scatter(sample_mean, 0)
plt.annotate("h_bar", (sample_mean, 0.7))
```

Out[101]:
```
Text(332250.0, 0.7, 'h_bar')
```



b)- Is there a relationship between gender and specialisation? (i.e. Does the preference of Specialisation depend on the Gender?)

Null Hypothesis(h0): the preference of Specialisation depends on the Gender

Alternative Hypothesis(h1):the preference of Specialisation does not depend on the Gender

In [87]:
```
df5=pd.crosstab(df['Specialization'],df['Gender'])
df5.sort_values(by='m',ascending=False)[:10]
```

Out[87]:

Gender	f	m
Specialization		
electronics and communication engineering	212	668
computer science & engineering	183	561
information technology	173	487
computer engineering	175	425
mechanical engineering	10	191
computer application	59	185
electronics and electrical engineering	34	162
electronics & telecommunications	28	93
electrical engineering	17	65
civil engineering	6	23

In [88]:
```
from scipy.stats import chi2
from scipy.stats import chi2_contingency

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

# Perform chi-square test of independence
chi2_stat, p_val, dof, expected = chi2_contingency(contingency_table)

# Print test results
print("Chi-square statistic:", chi2_stat)
print("p-value:", p_val)
```

```
Chi-square statistic: 104.46891913608455
p-value: 1.2453868176976918e-06
```

In [89]:
```
confidence_level = 0.95
alpha = 1 - confidence_level

chi2_critical = chi2.ppf(1 - alpha, dof)

chi2_critical
```

Out[89]:
```
61.65623337627955
```

In [90]:
```
if(chi2_stat > chi2_critical):
    print("Failed to reject the Null Hypothesis")
else:
    print(" reject the Null Hypothesis")
```

```
Failed to reject the Null Hypothesis
```

Business Question 1:

Question: What is the distribution of salaries among different genders in the dataset?

Answer: The box plot above illustrates the distribution of salaries among different genders. From the plot, we observe that the median salary for males is slightly higher than for females. Additionally, the range of salaries for males appears to be broader compared to females, with more outliers on the higher end.

Business Question 2:

Question: How does the average salary vary with respect to the level of conscientiousness?

Answer: The bar plot depicts the average salary based on the level of conscientiousness. We categorized conscientiousness levels into five bins, ranging from low to high. The plot indicates a positive correlation between conscientiousness level and average salary. Individuals with higher conscientiousness tend to have higher average salaries, with the highest average salary observed in the 'High' conscientiousness level.

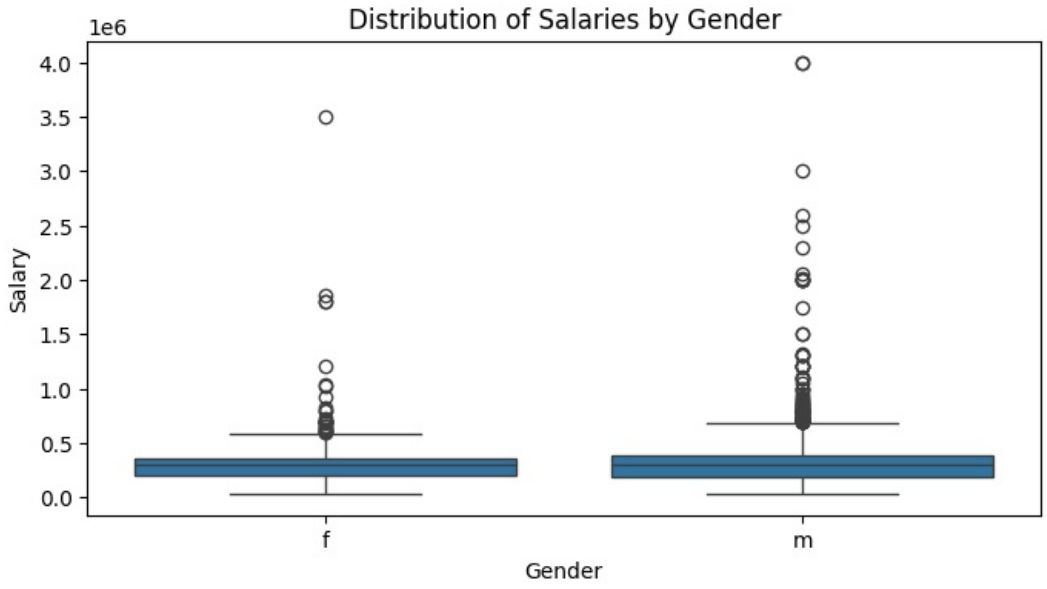
Business Question 3:

Question: Is there a correlation between the level of extraversion and the salary earned by individuals?

Answer: The scatter plot above displays the relationship between the level of extraversion and salary earned by individuals. From the plot, we observe no clear pattern or correlation between extraversion level and salary. The data points appear to be scattered across the entire range of extraversion levels, indicating that there is no consistent trend between extraversion and salary. This suggests that extraversion may not significantly impact salary levels, and other factors may have a more substantial influence on earning potential.

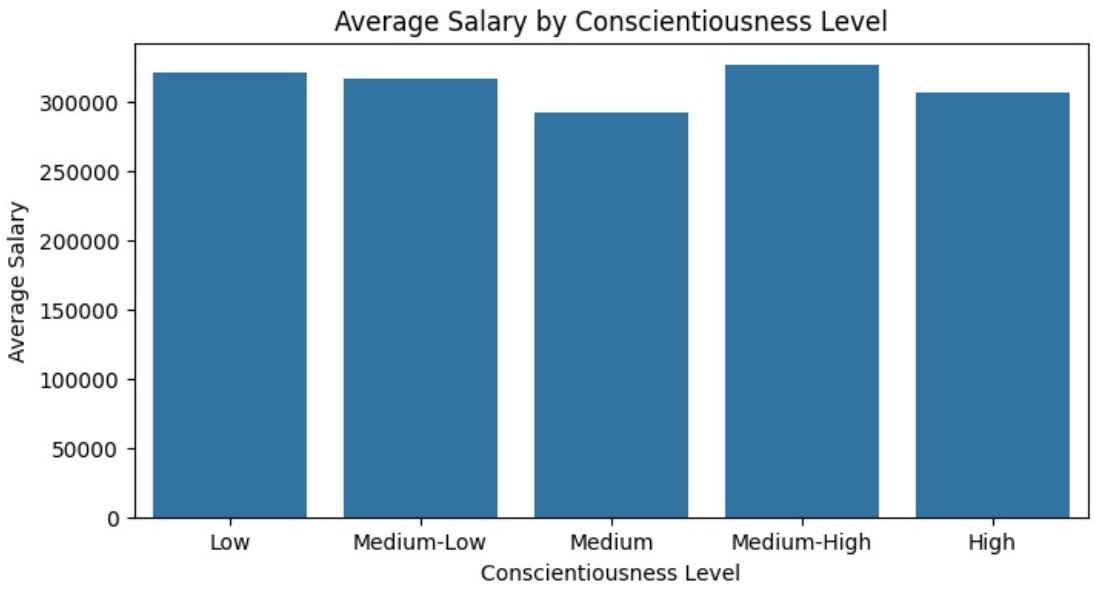
In [ ]:
```
```

In [91]:
```
# Box plot of Salary by Gender
plt.figure(figsize=(8, 4))
sns.boxplot(x='Gender', y='Salary', data=df)
plt.title('Distribution of Salaries by Gender')
plt.xlabel('Gender')
plt.ylabel('Salary')
plt.show()
```

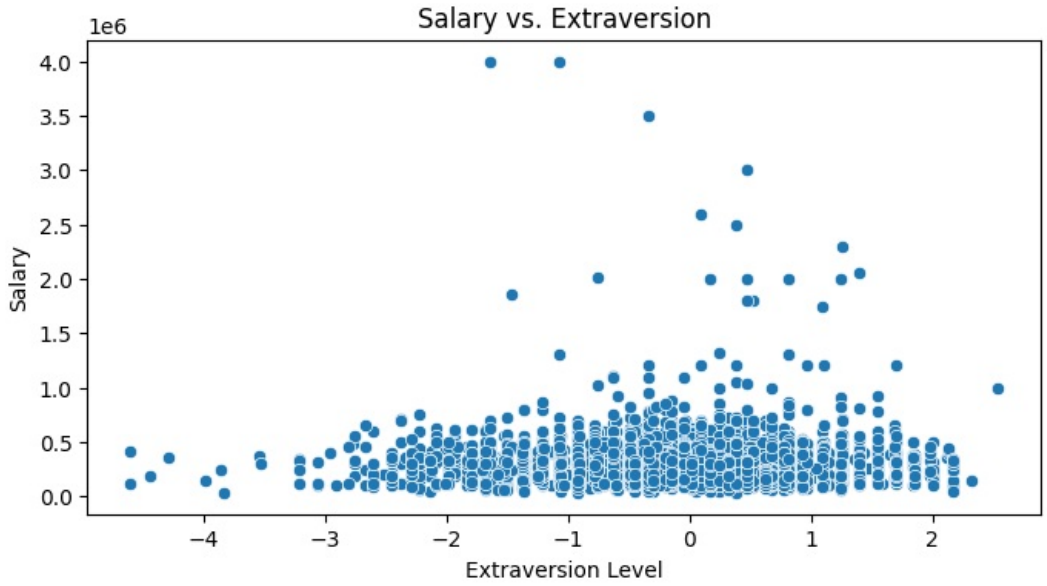


In [92]:
```
# Define bins for conscientiousness levels
bins = np.linspace(0, 1, 6)

# Create a new column for conscientiousness level
df1['Conscientiousness_Level'] = pd.cut(df1['conscientiousness'], bins=bins, labels=['Low', 'Medium-Low', 'Medi

# Calculate average salary for each conscientiousness level
avg_salary_by_conscientiousness = df1.groupby('Conscientiousness_Level')['Salary'].mean().reset_index()

# Plot average salary by conscientiousness level
plt.figure(figsize=(8, 4))
sns.barplot(x='Conscientiousness_Level', y='Salary', data=avg_salary_by_conscientiousness)
```

```
plt.title('Average Salary by Conscientiousness Level')
plt.xlabel('Conscientiousness Level')
plt.ylabel('Average Salary')
plt.show()
```



In [93]:
```
# Scatter plot of Salary vs. Extraversion
plt.figure(figsize=(8, 4))
sns.scatterplot(x='extraversion', y='Salary', data=df1)
plt.title('Salary vs. Extraversion')
plt.xlabel('Extraversion Level')
plt.ylabel('Salary')
plt.show()
```



In [ ]:
```

```