Exploratory Data Analysis on AMEO Dataset

Use Case Overview:

This study centers on individuals with engineering backgrounds, examining the employment outcomes of engineering graduates. The key dependent variables are salary, job titles, and job locations. The dataset also includes standardized scores in three areas: cognitive skills, technical skills, and personality traits, along with demographic details. With approximately 40 independent variables and 4,000 data points, the dataset contains both continuous and categorical attributes. Each candidate is uniquely identified by a specific ID.

Objective: Salary Prediction Model: Build a predictive model to estimate salaries based on academic achievements, standardized scores, and demographic data. Influential Factors Analysis: Identify the key factors that impact salary outcomes, including education background, skill levels, and work experience. Segmentation by Salary Bands: Develop a strategy to group candidates into salary bands, which could be used for targeted marketing and HR decisions.

```
import pandas as pd
from datetime import datetime, timedelta
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

```
df=pd.read_csv("/content/sample_data/data.xlsx - Sheet1.csv")
```

```
df.head()
```

	Unnamed: 0	Id	Salary	Doj	Dol	Designation	Jobcity	Gender	Dob	10Percentage	...	Computerscience
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

```
df.tail()
```

	Unnamed: 0	Id	Salary	Doj	Dol	Designation	Jobcity	Gender	Dob	10Percentage	...	Compute
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/15/87 0:00	52.09	...	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/27/92 0:00	90.00	...	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	...	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	...	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	...	

5 rows × 39 columns

```
df.shape
```

```
(3998, 39)
```

```
df.size
```

```
155922
```

```
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
```

```
df.isnull().sum()
```

	0
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

No Missing Values in the data set

```
df[df.duplicated(keep=False)]
```

Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	ComputerScience	MechanicalEı

0 rows × 39 columns

Data Cleaning

Standardization of Column Names:

Ensuring Correct Data Formats:

Conversion of Data Types:

Titlecase Application:

Handling Missing Data:

Outlier Detection and Treatment:

Duplicate Entry Check:

```
df.columns = [col.title() for col in df.columns]

column_mapping = {'Doj': 'DOJ', 'Dol': 'DOL', 'Dob': 'DOB'}
df.rename(columns=column_mapping, inplace=True)
```

```
df
```

	Unnamed: 0	Id	Salary	Doj	Dol	Designation	Jobcity	Gender	Dob	10Percentage	...	Comput
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.30	...	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.40	...	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.00	...	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.60	...	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.00	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/15/87 0:00	52.09	...	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/27/92 0:00	90.00	...	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	...	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	...	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	...	

3998 rows × 39 columns

```
df['Salary'] = pd.to_numeric(df['Salary'], errors='coerce')
```

```
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
```

```
def titlecase_column(column_name):
    df[column_name] = df[column_name].astype(str).str.title()

categorical_columns = df.select_dtypes(include='object').columns
for column in categorical_columns:
    titlecase_column(column)

# Display the DataFrame with titlecased contents
print(df)
```

```
3994      Train  752781   100000.0 2013-07-01   7/1/13 0:00
3995      Train  355888   320000.0 2013-07-01    2024-10-03
3996      Train  947111   200000.0 2014-07-01   1/1/15 0:00
3997      Train  324966   400000.0 2013-02-01    2024-10-03

                      Designation           Jobcity Gender        DOB  \
0         Senior Quality Engineer         Bangalore      F 1990-02-19   
1               Assistant Manager            Indore      M 1989-10-04   
2                Systems Engineer           Chennai      F 1992-08-03   
3        Senior Software Engineer           Gurgaon      M 1989-12-05   
4                             Get           Manesar      M 1991-02-27   
...                           ...               ...    ...        ...   
3993            Software Engineer        New Delhi      M 1987-04-15   
3994             Technical Writer         Hyderabad      F 1992-08-27   
3995  Associate Software Engineer         Bangalore      M 1991-07-03   
3996           Software Developer  Asifabadbanglore      F 1992-03-20   
3997      Senior Systems Engineer           Chennai      F 1991-02-26   

      10Percentage  ... Computerscience  Mechanicalengg  Electricalengg  \
0            84.30  ...              -1              -1              -1   
```

```
3996         78.72  ...             438             -1             -1
3997         70.60  ...              -1             -1             -1

     Telecomengg  Civilengg  Conscientiousness Agreeableness Extraversion  \
0             -1         -1             0.9737        0.8128       0.5269
1             -1         -1            -0.7335        0.3789       1.2396
2             -1         -1             0.2718        1.7109       0.1637
3             -1         -1             0.0464        0.3448      -0.3440
4             -1         -1            -0.8810       -0.2793      -1.0697
...          ...        ...                ...           ...          ...
3993          -1         -1            -0.1082        0.3448       0.2366
3994          -1         -1            -0.3027        0.8784       0.9322
3995          -1         -1            -1.5765       -1.5273      -1.5051
3996          -1         -1            -0.1590        0.0459      -0.4511
3997          -1         -1            -1.1128       -0.2793      -0.6343

      Nueroticism  Openess_To_Experience
0         1.35490                -0.4455
1        -0.10760                 0.8637
2        -0.86820                 0.6721
3        -0.40780                -0.9194
4         0.09163                -0.1295
...           ...                    ...
3993      0.64980                -0.9194
3994      0.77980                -0.0943
3995     -1.31840                -0.7615
3996     -0.36120                -0.0943
3997      1.32553                -0.6035

[3998 rows x 39 columns]
```

```
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   datetime64[ns]
 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   datetime64[ns]
 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: datetime64[ns](2), float64(10), int64(17), object(10)
memory usage: 1.2+ MB
```

Numerical Analysis

```
num_analysis=df.select_dtypes(include=['int64','float'])
cat_analysis=df.select_dtypes(include=['object'])
```

```
for col in num_analysis:
    print("_"*10,col,'_'*10)
    print(num_analysis[col].agg(['min','mean','median','max']))
#     print(num_analysis[col].mean())
```

```
mean       90.742371
median     -1.000000
max       715.000000
Name: Computerscience, dtype: float64
__________ Mechanicalengg __________
min        -1.000000
mean       22.974737
median     -1.000000
max       623.000000
Name: Mechanicalengg, dtype: float64
__________ Electricalengg __________
min        -1.000000
mean       16.478739
median     -1.000000
max       676.000000
Name: Electricalengg, dtype: float64
__________ Telecomengg __________
min        -1.000000
mean       31.851176
median     -1.000000
max       548.000000
Name: Telecomengg, dtype: float64
__________ Civilengg __________
min        -1.000000
mean        2.683842
median     -1.000000
max       516.000000
Name: Civilengg, dtype: float64
__________ Conscientiousness __________
min      -4.126700
mean     -0.037831
median    0.046400
max       1.995300
Name: Conscientiousness, dtype: float64
__________ Agreeableness __________
min      -5.781600
mean      0.146496
median    0.212400
max       1.904800
Name: Agreeableness, dtype: float64
__________ Extraversion __________
min      -4.600900
mean      0.002763
median    0.091400
max       2.535400
Name: Extraversion, dtype: float64
__________ Nueroticism __________
min      -2.643000
mean     -0.169033
median   -0.234400
max       3.352500
Name: Nueroticism, dtype: float64
__________ Openess_To_Experience __________
min      -7.37570
mean     -0.13811
median   -0.09430
max       1.82240
Name: Openess_To_Experience, dtype: float64
```

Categorical Analysis

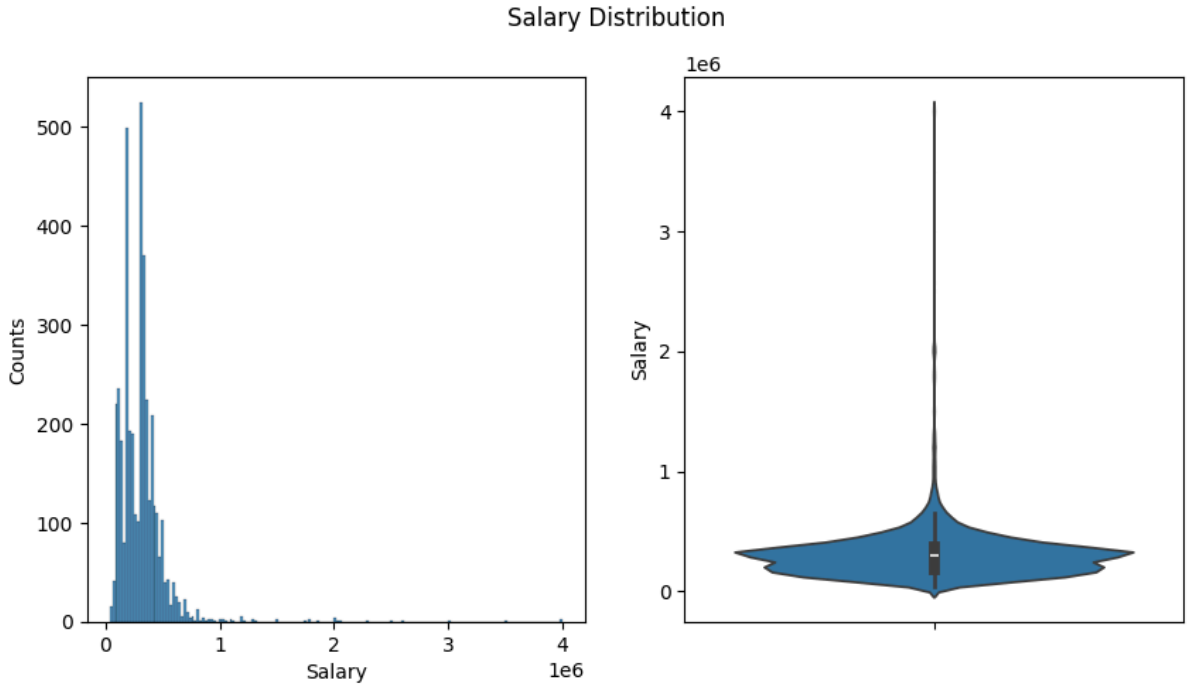
```
for col in cat_analysis:
    print("_"*10,col,"_"*10)
    print(cat_analysis[col].agg(['count','unique','nunique']))
    print(cat_analysis[col].value_counts())
```

```
Electronics And Computer Engineering             3
Electrical And Power Engineering                 2
Biomedical Engineering                           2
Information & Communication Technology           2
Industrial Engineering                           2
Computer Science                                 2
Metallurgical Engineering                        2
Power Systems And Automation                     1
Control And Instrumentation Engineering          1
Mechanical & Production Engineering              1
Embedded Systems Technology                      1
Polymer Technology                               1
Computer And Communication Engineering           1
Information Science                              1
Internal Combustion Engine                       1
Computer Networking                              1
Ceramic Engineering                              1
Electronics                                      1
Industrial & Management Engineering              1
Name: count, dtype: int64
__________ Collegestate __________
count                                                 3998
unique     [Andhra Pradesh, Madhya Pradesh, Uttar Pradesh...
nunique                                                 26
Name: Collegestate, dtype: object
Collegestate
Uttar Pradesh        915
Karnataka            370
Tamil Nadu           367
Telangana            319
Maharashtra          262
Andhra Pradesh       225
West Bengal          196
Punjab               193
Madhya Pradesh       189
Haryana              180
Rajasthan            174
Orissa               172
Delhi                162
Uttarakhand          113
Kerala                33
Jharkhand             28
Chhattisgarh          27
Gujarat               24
Himachal Pradesh      16
Bihar                 10
Jammu And Kashmir      7
Assam                  5
Union Territory        5
Sikkim                 3
Meghalaya              2
Goa                    1
Name: count, dtype: int64
```

Univariate Analysis

```
fig,axs=plt.subplots(1,2,figsize=(10,5))
plt.suptitle('Salary Distribution')
sns.histplot(df['Salary'],ax=axs[0])
sns.violinplot(df['Salary'],ax=axs[1])
axs[1].set_ylabel('Salary')
axs[0].set_xlabel("Salary")
axs[0].set_ylabel('Counts')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 list-l
  data_subset = grouped_data.get_group(pd_key)
```




```
discrete_df = df.select_dtypes(include=['object'])

numerical_df = df.select_dtypes(include=['int64', 'float64'])


def discrete_univariate_analysis(discrete_data):
    for col_name in discrete_data:
        print("*"*10, col_name, "*"*10)
        print(discrete_data[col_name].agg(['count', 'nunique', 'unique']))
        print('Value Counts: \n', discrete_data[col_name].value_counts())
        print()


discrete_univariate_analysis(discrete_df)
```

```
Karnataka             370
Tamil Nadu            367
Telangana             319
Maharashtra           262
Andhra Pradesh        225
West Bengal           196
Punjab                193
Madhya Pradesh        189
Haryana               180
Rajasthan             174
Orissa                172
Delhi                 162
Uttarakhand           113
Kerala                 33
Jharkhand              28
Chhattisgarh           27
Gujarat                24
Himachal Pradesh       16
Bihar                  10
Jammu And Kashmir       7
Assam                   5
Union Territory         5
Sikkim                  3
Meghalaya               2
Goa                     1
Name: count, dtype: int64
```

```
def numerical_univariate_analysis(numerical_data):
    for col_name in numerical_data:
        print("*"*10, col_name, "*"*10)
        print(numerical_data[col_name].agg(['min', 'max', 'mean', 'median', 'std']))
        print()
```

```
numerical_univariate_analysis(numerical_df)
```

```
mean     -0.169033
median   -0.234400
std       1.007580
Name: Nueroticism, dtype: float64

********** Openess_To_Experience **********
min      -7.375700
max       1.822400
mean     -0.138110
median   -0.094300
std       1.008075
Name: Openess_To_Experience, dtype: float64
```

```
numerical_df.columns
```

```
Index(['Id', 'Salary', '10Percentage', '12Graduation', '12Percentage',
       'Collegeid', 'Collegetier', 'Collegegpa', 'Collegecityid',
       'Collegecitytier', 'Graduationyear', 'English', 'Logical', 'Quant',
       'Domain', 'Computerprogramming', 'Electronicsandsemicon',
       'Computerscience', 'Mechanicalengg', 'Electricalengg', 'Telecomengg',
       'Civilengg', 'Conscientiousness', 'Agreeableness', 'Extraversion',
       'Nueroticism', 'Openess_To_Experience'],
      dtype='object')
```

```
discrete_num_cols=['Id','12Graduation','Graduationyear','Collegecityid', 'Collegeid','Collegecitytier','Collegetier',
       'Domain', 'Computerprogramming', 'Electronicsandsemicon',
       'Computerscience', 'Mechanicalengg', 'Electricalengg', 'Telecomengg',
       'Civilengg', 'Conscientiousness', 'Agreeableness', 'Extraversion',
       'Nueroticism', 'Openess_To_Experience']
```

```
numerical_df.drop(columns=discrete_num_cols, axis=1, inplace=True)
```

```
print('Shape:', numerical_df.shape)
print('Columns:', numerical_df.columns)
```

```
Shape: (3998, 4)
Columns: Index(['Salary', '10Percentage', '12Percentage', 'Collegegpa'], dtype='object')
```

```
numerical_df
```

	Salary	10Percentage	12Percentage	Collegegpa
0	420000.0	84.30	95.80	78.00
1	500000.0	85.40	85.00	70.06
2	325000.0	85.00	68.20	70.00
3	1100000.0	85.60	83.60	74.64
4	200000.0	78.00	76.80	73.90
...	...	...	...	...
3993	280000.0	52.09	55.50	61.50
3994	100000.0	90.00	93.00	77.30
3995	320000.0	81.86	65.50	70.00
3996	200000.0	78.72	69.88	70.42
3997	400000.0	70.60	68.00	68.00

3998 rows × 4 columns

Next steps: Generate code with numerical_df | View recommended plots | New interactive sheet

```
numerical_univariate_analysis(numerical_df)
```

```
********** Salary **********
min       3.500000e+04
max       4.000000e+06
mean      3.076998e+05
median    3.000000e+05
std       2.127375e+05
Name: Salary, dtype: float64

********** 10Percentage **********
min       43.000000
max       97.760000
mean      77.925443
median    79.150000
std        9.850162
Name: 10Percentage, dtype: float64

********** 12Percentage **********
min       40.000000
max       98.700000
mean      74.466366
median    74.400000
std       10.999933
Name: 12Percentage, dtype: float64

********** Collegegpa **********
min        6.450000
max       99.930000
mean      71.486171
median    71.720000
std        8.167338
Name: Collegegpa, dtype: float64
```

```
discrete_univariate_analysis(discrete_df)
```

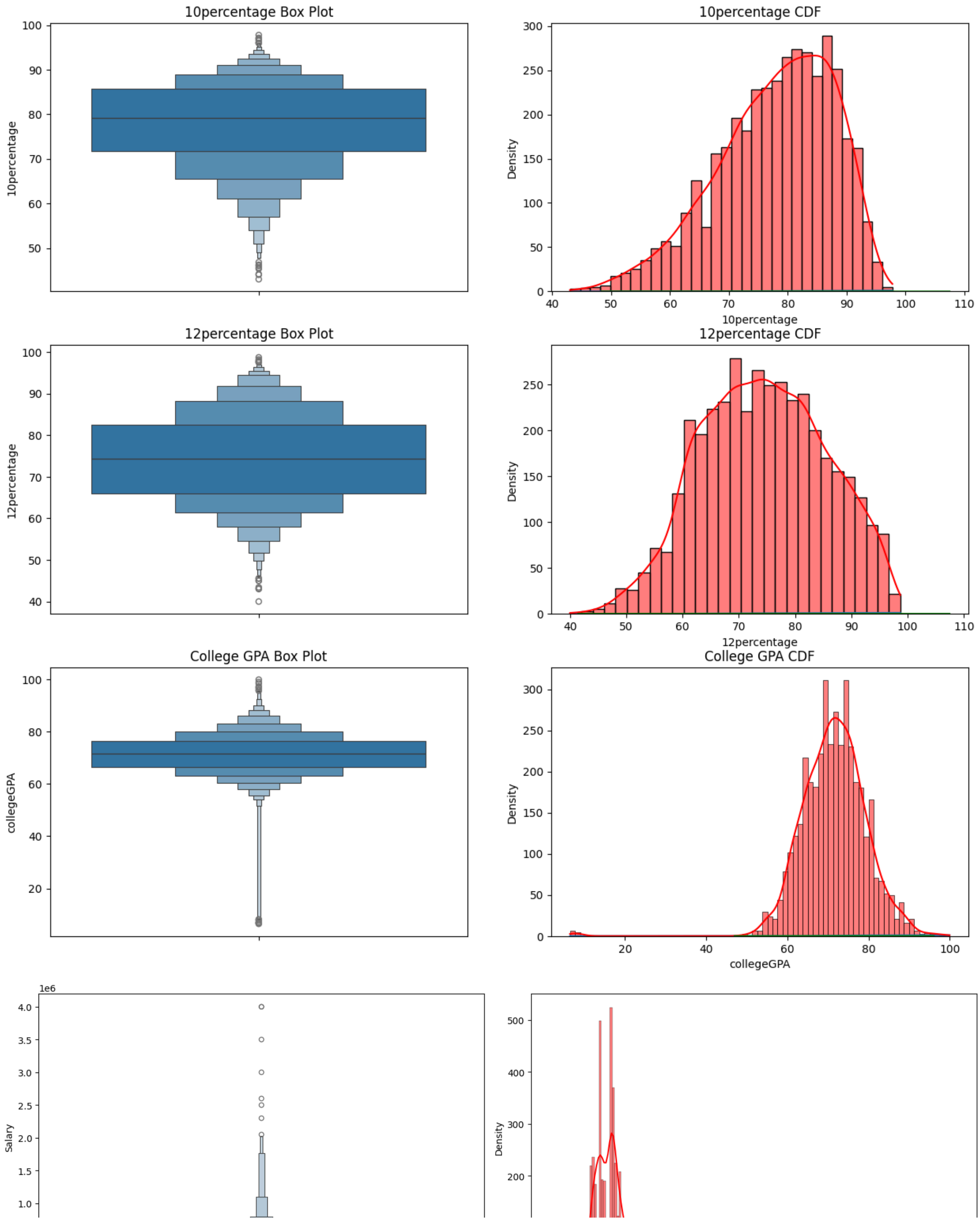
```
Himachal Pradesh      16
Bihar                 10
Jammu And Kashmir      7
Assam                  5
Union Territory        5
Sikkim                 3
Meghalaya              2
Goa                    1
Name: count, dtype: int64
```

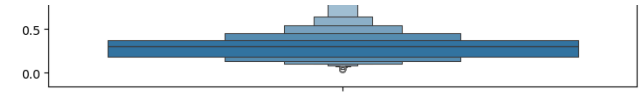
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm

# Generate normal distribution parameters
mu_10percentage, std_10percentage = np.mean(df['10percentage']), np.std(df['10percentage'])
mu_12percentage, std_12percentage = np.mean(df['12percentage']), np.std(df['12percentage'])
mu_collegeGPA, std_collegeGPA = np.mean(df['collegeGPA']), np.std(df['collegeGPA'])
mu_Salary, std_Salary = np.mean(df['Salary']), np.std(df['Salary'])
mu_conscientiousness, std_conscientiousness = np.mean(df['conscientiousness']), np.std(df['conscientiousness'])
mu_nueroticism, std_nueroticism = np.mean(df['nueroticism']), np.std(df['nueroticism'])
mu_openess_to_experience, std_openess_to_experience = np.mean(df['openess_to_experience']), np.std(df['openess_to_expe

# First Frame: 10percentage, 12percentage, collegeGPA
fig, axs1 = plt.subplots(3, 2, figsize=(15, 15))
plt.suptitle('Numerical Univariate Analysis')

# 10percentage Distribution
sns.boxenplot(df['10percentage'], ax=axs1[0, 0])
sns.histplot(df['10percentage'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs1[0, 1])
sns.histplot(df['10percentage'], kde=True, color='red', ax=axs1[0, 1], label='KDE')

x_10percentage = np.linspace(mu_10percentage - 3*std_10percentage, mu_10percentage + 3*std_10percentage, 100)
axs1[0, 1].plot(x_10percentage, norm.pdf(x_10percentage, mu_10percentage, std_10percentage), color='green', label='Nor

# 12percentage Distribution
sns.boxenplot(df['12percentage'], ax=axs1[1, 0])
sns.histplot(df['12percentage'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs1[1, 1])
sns.histplot(df['12percentage'], kde=True, color='red', ax=axs1[1, 1], label='KDE')

x_12percentage = np.linspace(mu_12percentage - 3*std_12percentage, mu_12percentage + 3*std_12percentage, 100)
axs1[1, 1].plot(x_12percentage, norm.pdf(x_12percentage, mu_12percentage, std_12percentage), color='green', label='Nor

# collegeGPA Distribution
sns.boxenplot(df['collegeGPA'], ax=axs1[2, 0])
sns.histplot(df['collegeGPA'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs1[2, 1])
sns.histplot(df['collegeGPA'], kde=True, color='red', ax=axs1[2, 1], label='KDE')

x_collegeGPA = np.linspace(mu_collegeGPA - 3*std_collegeGPA, mu_collegeGPA + 3*std_collegeGPA, 100)
axs1[2, 1].plot(x_collegeGPA, norm.pdf(x_collegeGPA, mu_collegeGPA, std_collegeGPA), color='green', label='Normal Dist

# Set titles and labels
axs1[0, 0].set_title('10percentage Box Plot')
axs1[0, 1].set_title('10percentage CDF')
axs1[1, 0].set_title('12percentage Box Plot')
axs1[1, 1].set_title('12percentage CDF')
axs1[2, 0].set_title('College GPA Box Plot')
axs1[2, 1].set_title('College GPA CDF')

# Second Frame: Salary, Conscientiousness, Nueroticism, openess_to_experience
fig, axs2 = plt.subplots(4, 2, figsize=(15, 20))

# Salary Distribution
sns.boxenplot(df['Salary'], ax=axs2[0, 0])
sns.histplot(df['Salary'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs2[0, 1])
sns.histplot(df['Salary'], kde=True, color='red', ax=axs2[0, 1], label='KDE')

x_Salary = np.linspace(mu_Salary - 3*std_Salary, mu_Salary + 3*std_Salary, 100)
axs2[0, 1].plot(x_Salary, norm.pdf(x_Salary, mu_Salary, std_Salary), color='green', label='Normal Distribution')

# Conscientiousness Distribution
sns.boxenplot(df['conscientiousness'], ax=axs2[1, 0])
sns.histplot(df['conscientiousness'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs2[1, 1])
sns.histplot(df['conscientiousness'], kde=True, color='red', ax=axs2[1, 1], label='KDE')

x_conscientiousness = np.linspace(mu_conscientiousness - 3*std_conscientiousness, mu_conscientiousness + 3*std_conscie
axs2[1, 1].plot(x_conscientiousness, norm.pdf(x_conscientiousness, mu_conscientiousness, std_conscientiousness), color
```

```
axs2[1, 1].plot(x_conscientiousness, norm.pdf(x_conscientiousness, mu_conscientiousness, std_conscientiousness), color=

# Nueroticism Distribution
sns.boxenplot(df['nueroticism'], ax=axs2[2, 0])
sns.histplot(df['nueroticism'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs2[2, 1])
sns.histplot(df['nueroticism'], kde=True, color='red', ax=axs2[2, 1], label='KDE')

x_nueroticism = np.linspace(mu_nueroticism - 3*std_nueroticism, mu_nueroticism + 3*std_nueroticism, 100)
axs2[2, 1].plot(x_nueroticism, norm.pdf(x_nueroticism, mu_nueroticism, std_nueroticism), color='green', label='Normal

# Openness to Experience Distribution
sns.boxenplot(df['openess_to_experience'], ax=axs2[3, 0])
sns.histplot(df['openess_to_experience'], kde=True, cumulative=True, stat="density", common_norm=False, ax=axs2[3, 1])
sns.histplot(df['openess_to_experience'], kde=True, color='red', ax=axs2[3, 1], label='KDE')

x_openess_to_experience = np.linspace(mu_openess_to_experience - 3*std_openess_to_experience, mu_openess_to_experience
axs2[3, 1].plot(x_openess_to_experience, norm.pdf(x_openess_to_experience, mu_openess_to_experience, std_openess_to_ex

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

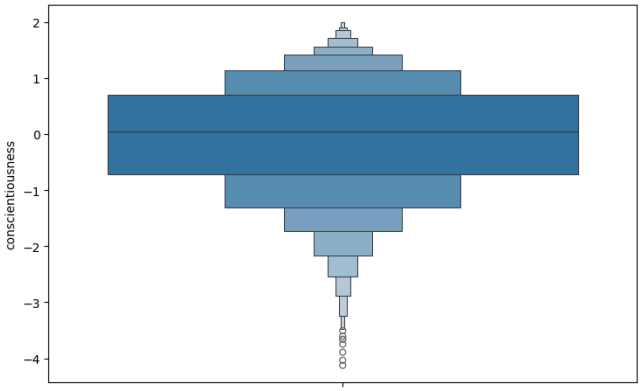
```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949: FutureWarning: When grouping with a length-1 li
  data_subset = grouped_data.get_group(pd_key)
```

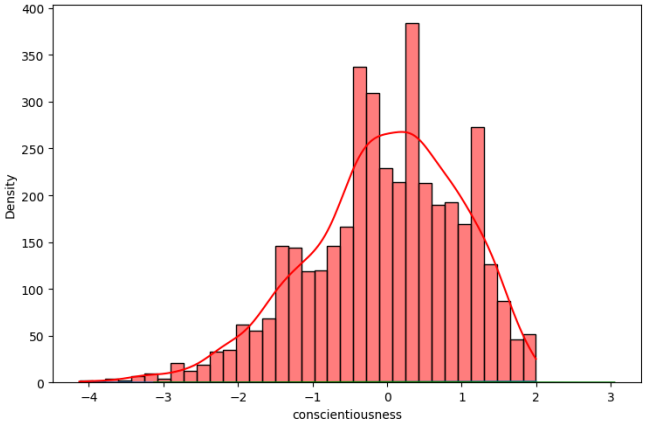
Numerical Univariate Analysis

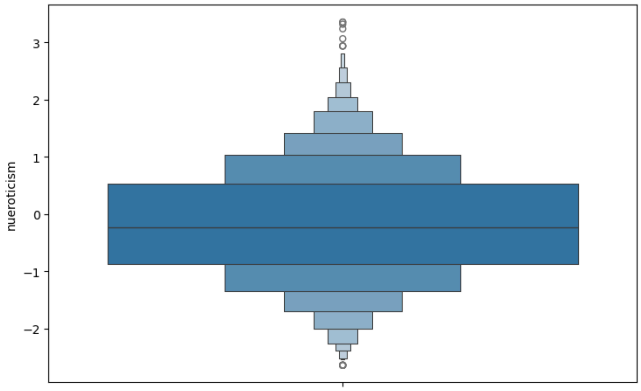


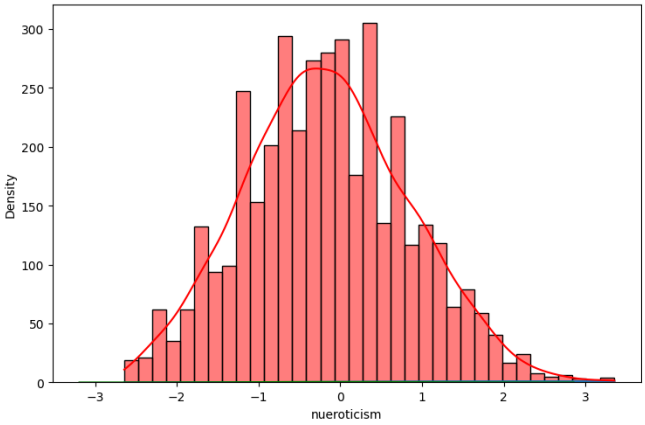

0.5
0.0

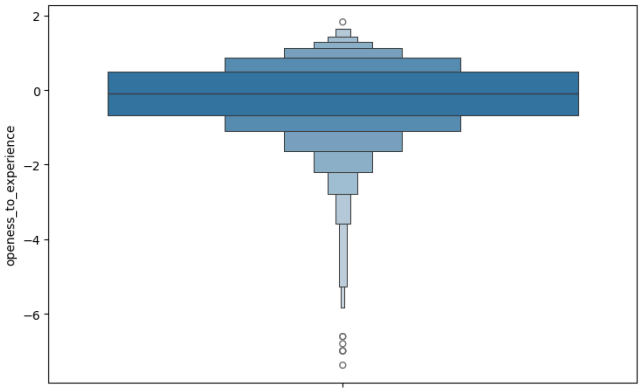

100
0
0
1
2
3
4
Salary
1e6

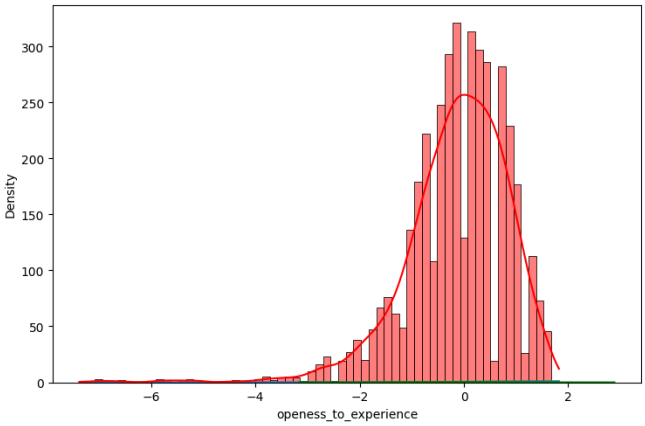

conscientiousness
2
1
0
−1
−2
−3
−4


Density
400
350
300
250
200
150
100
50
0
−4
−3
−2
−1
0
1
2
3
conscientiousness

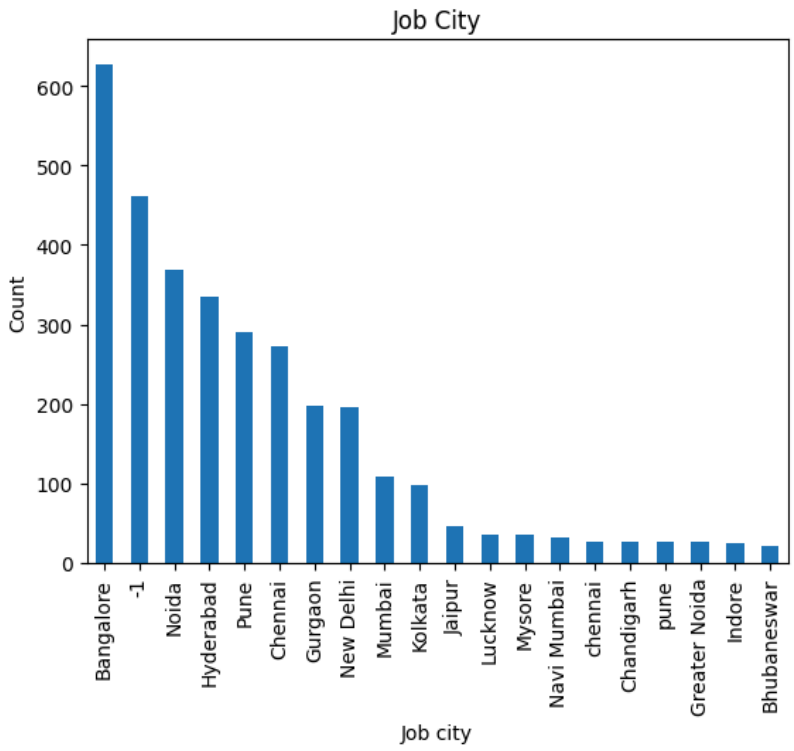

nueroticism
3
2
1
0
−1
−2


Density
300
250
200
150
100
50
0
−3
−2
−1
0
1
2
3
nueroticism


openess_to_experience
2
0
−2
−4
−6


Density
300
250
200
150
100
50
0
−6
−4
−2
0
2
openess_to_experience

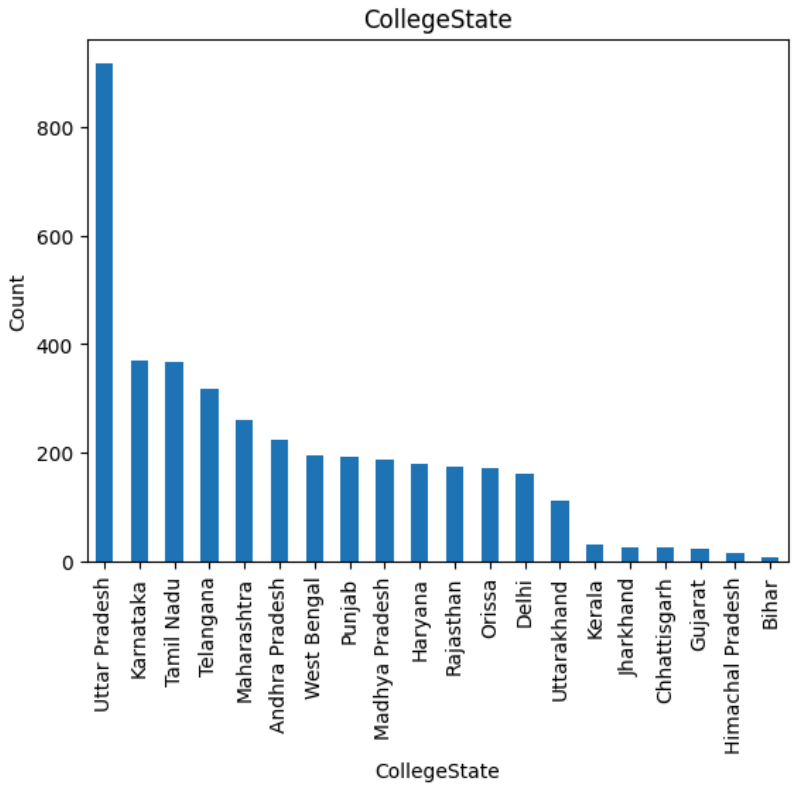
```
df['JobCity'].value_counts().head(20).plot(kind='bar')
plt.title('Job City')
plt.ylabel('Count')
plt.xlabel('Job city')
plt.show()
```




```
import numpy as np
import scipy.stats as stats

def binomial_distribution_generator(n, p, size):
    return np.random.binomial(n=n, p=p, size=size)

def discrete_non_viz_analysis(data):
    # Analysis code for non-visual analysis
    pass

def discrete_viz_analysis(data):
    # Analysis code for visual analysis
    pass


jobcity_data = [city for city, count in df['JobCity'].value_counts().items() for _ in range(count)]

binomial_dist = binomial_distribution_generator(n=10, p=0.30, size=len(jobcity_data))


n_trials = 10
prob_success = 0.30
var = stats.binom.rvs(n=n_trials, p=prob_success, size=len(jobcity_data))

# Computing PMF and CDF
print("Probability of 0 success out of 10 trials:", stats.binom.pmf(k=0, n=n_trials, p=prob_success))
print("Probability of 1 success out of 10 trials:", stats.binom.pmf(k=1, n=n_trials, p=prob_success))
print("Probability of 2 successes out of 10 trials:", stats.binom.pmf(k=2, n=n_trials, p=prob_success))
print("Probability of <= 2 success out of 10 trials:", stats.binom.cdf(k=2, n=n_trials, p=prob_success))
```

```
Probability of 0 success out of 10 trials: 0.0282475249
Probability of 1 success out of 10 trials: 0.12106082099999989
Probability of 2 successes out of 10 trials: 0.2334744405000001
Probability of <= 2 success out of 10 trials: 0.3827827863999998
```

```
df['CollegeState'].value_counts().head(20).plot(kind='bar')
plt.title('CollegeState')
plt.ylabel('Count')
plt.xlabel('CollegeState')
plt.show()
```




```
df.columns
```

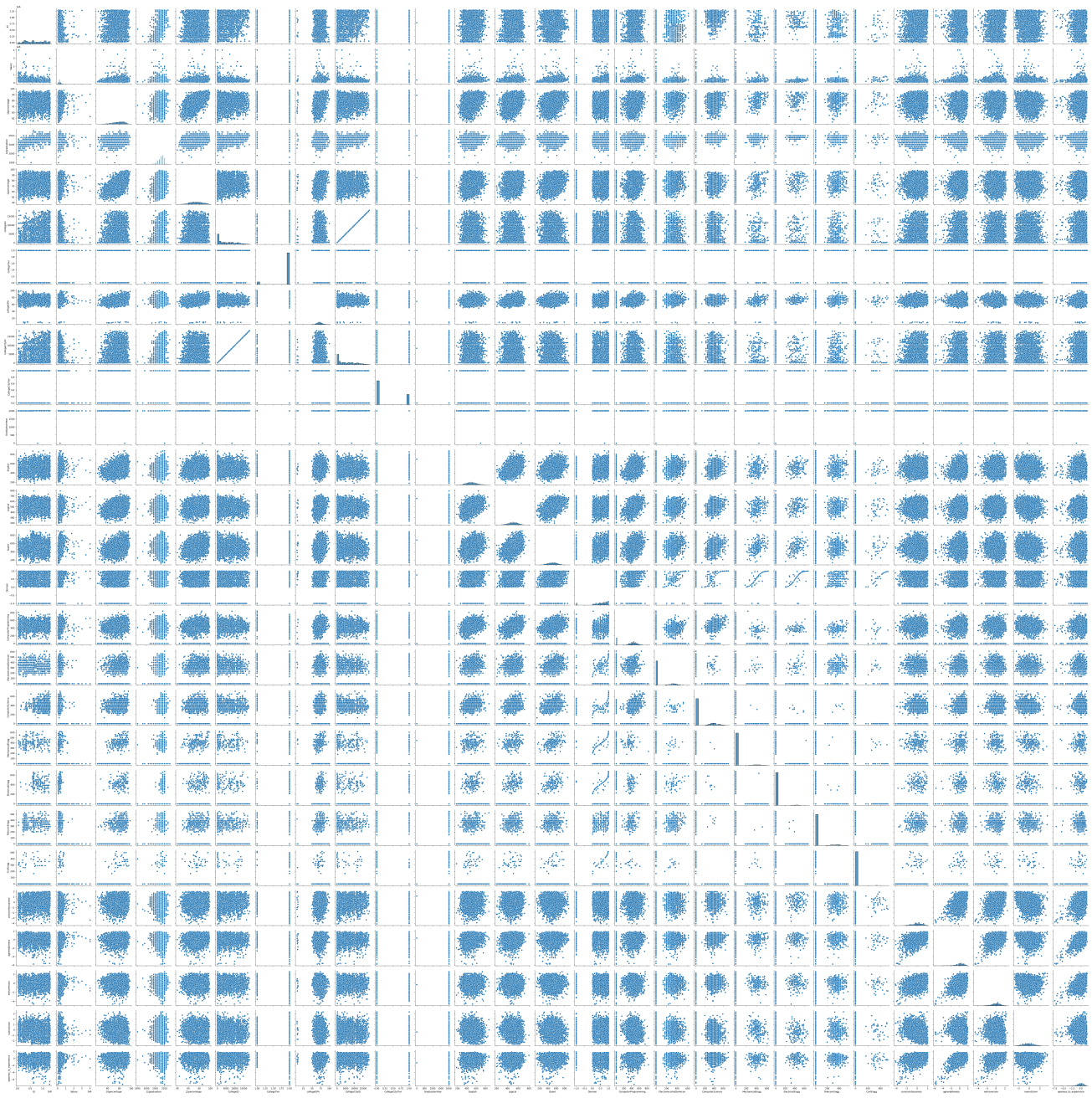
```
Index(['Unnamed: 0', 'ID', 'Salary', 'DOJ', 'DOL', 'Designation', 'JobCity',
       'Gender', 'DOB', '10percentage', '10board', '12graduation',
       '12percentage', '12board', 'CollegeID', 'CollegeTier', 'Degree',
       'Specialization', 'collegeGPA', 'CollegeCityID', 'CollegeCityTier',
     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

```
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
```

```
sns.pairplot(df)
plt.show()
```



```
numerical_df.columns
```

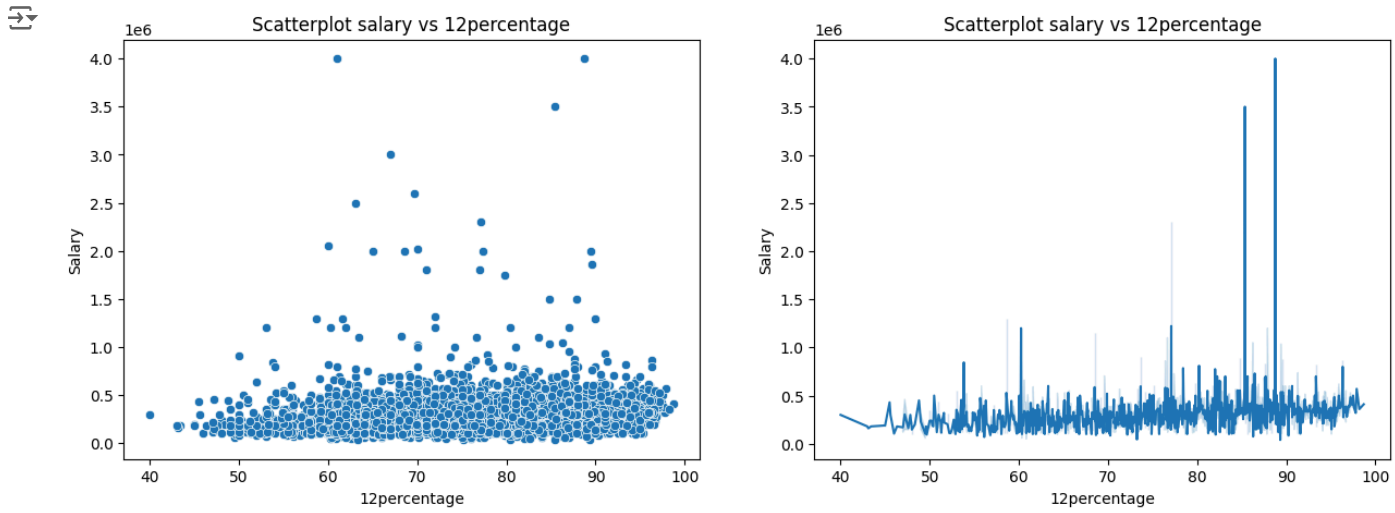
```
Index(['Salary', '10Percentage', '12Percentage', 'Collegegpa'], dtype='object')
```

Bi-Variate Analysis

What is the Relationship between Salary and 12percentage percentage?

```
import scipy.stats as stats

def pearson_test(data1, data2, significance_level):
    correlation_coefficient, p_value = stats.pearsonr(data1, data2)
    print('correlation coefficient=%.3f, p-value=%.3f' % (correlation_coefficient, p_value))

    if p_value < significance_level:
        print("Reject null hypothesis: There is significant correlation.")
    else:
        print("Fail to reject null hypothesis: There is no significant correlation.")
```

```
fig,axs=plt.subplots(1,2,figsize=(15,5),)
sns.scatterplot(data=df,x='12percentage',y='Salary',ax=axs[0])
axs[0].set_title('Scatterplot salary vs 12percentage')
sns.lineplot(data=df,x='12percentage',y='Salary',ax=axs[1])
axs[1].set_title('Scatterplot salary vs 12percentage')

plt.show()
```



12th Percentage vs. Salary:

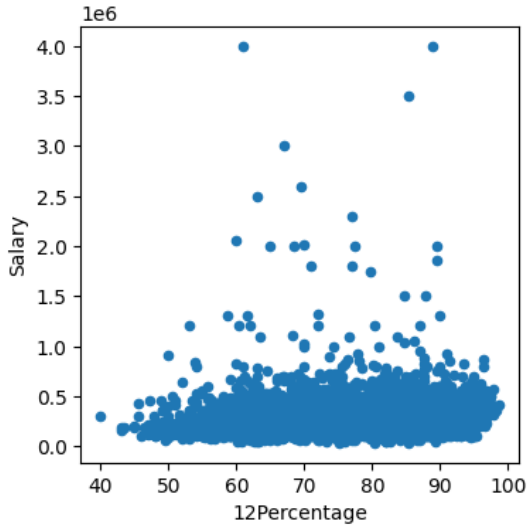
Correlation Analysis: There is a notable positive correlation between 12th percentage and salary (correlation coefficient = 0.169, p-value = 0.000), indicating that higher 12th percentages are linked to higher salaries.

Impact on Salary Negotiation: Candidates with higher 12th percentages may have stronger leverage when negotiating salaries during job offers or promotions.

Employer Consideration: Employers may consider strong 12th-grade performance as a reflection of academic diligence and potential contribution, influencing salary decisions.

```
df.plot(kind='scatter', x='12Percentage', y='Salary', figsize=(4, 4))
```

```
<Axes: xlabel='12Percentage', ylabel='Salary'>
```




```
pd.crosstab(df['Specialization'],df['Designation'])
```

Designation	.net developer	.net web developer	account executive	account manager	admin assistant	administrative coordinator	administrative support	airc techni
Specialization								
aeronautical engineering	0	0	0	0	0	0	0	
applied electronics and instrumentation	0	0	0	0	0	0	0	
automobile/automotive engineering	0	0	0	0	0	0	0	
biomedical engineering	0	0	0	0	0	0	0	
biotechnology	0	0	0	0	0	0	0	
ceramic engineering	0	0	0	0	0	0	0	
chemical engineering	0	0	0	0	0	0	0	
civil engineering	0	0	0	0	0	0	0	
computer and communication engineering	0	0	0	0	0	0	0	
computer application	4	1	0	0	0	0	1	
computer engineering	7	0	0	1	0	0	0	
computer networking	0	0	0	0	0	0	0	
computer science	0	0	0	0	0	0	0	
computer science & engineering	8	2	1	0	1	1	0	
computer science and technology	0	0	0	0	0	0	0	
control and instrumentation engineering	0	0	0	0	0	0	0	
electrical and power engineering	0	0	0	0	0	0	0	
electrical engineering	0	0	0	0	0	0	0	
electronics	0	0	0	0	0	0	0	
electronics & instrumentation eng	0	0	0	0	0	0	0	
electronics & telecommunications	0	0	1	0	0	0	0	
electronics and communication engineering	6	0	1	0	0	0	0	
electronics and computer engineering	0	0	0	0	0	0	0	
electronics and electrical engineering	0	0	0	0	0	0	0	
electronics and instrumentation engineering	0	0	0	0	0	0	0	
electronics engineering	0	0	0	0	0	0	0	
embedded systems technology	0	0	0	0	0	0	0	
industrial & management engineering	0	0	0	0	0	0	0	
industrial & production engineering	0	0	0	0	0	0	0	
industrial engineering	0	0	0	0	0	0	0	

information & communication technology	0	0	0	0	0	0	0
information science	0	0	0	0	0	0	0
information science engineering	1	0	0	0	0	0	0
information technology	8	1	1	0	1	0	0
instrumentation and control engineering	0	0	0	0	0	0	0
instrumentation engineering	0	0	0	0	0	0	0
internal combustion engine	0	0	0	0	0	0	0
mechanical & production engineering	0	0	0	0	0	0	0
mechanical and automation	0	0	0	0	0	0	0
mechanical engineering	0	0	0	0	0	0	0
mechatronics	0	0	0	0	0	0	0
metallurgical engineering	0	0	0	0	0	0	0
other	0	0	0	0	0	0	0
polymer technology	0	0	0	0	0	0	0
power systems and automation	0	0	0	0	0	0	0
telecommunication engineering	0	0	0	0	0	0	0

46 rows × 419 columns

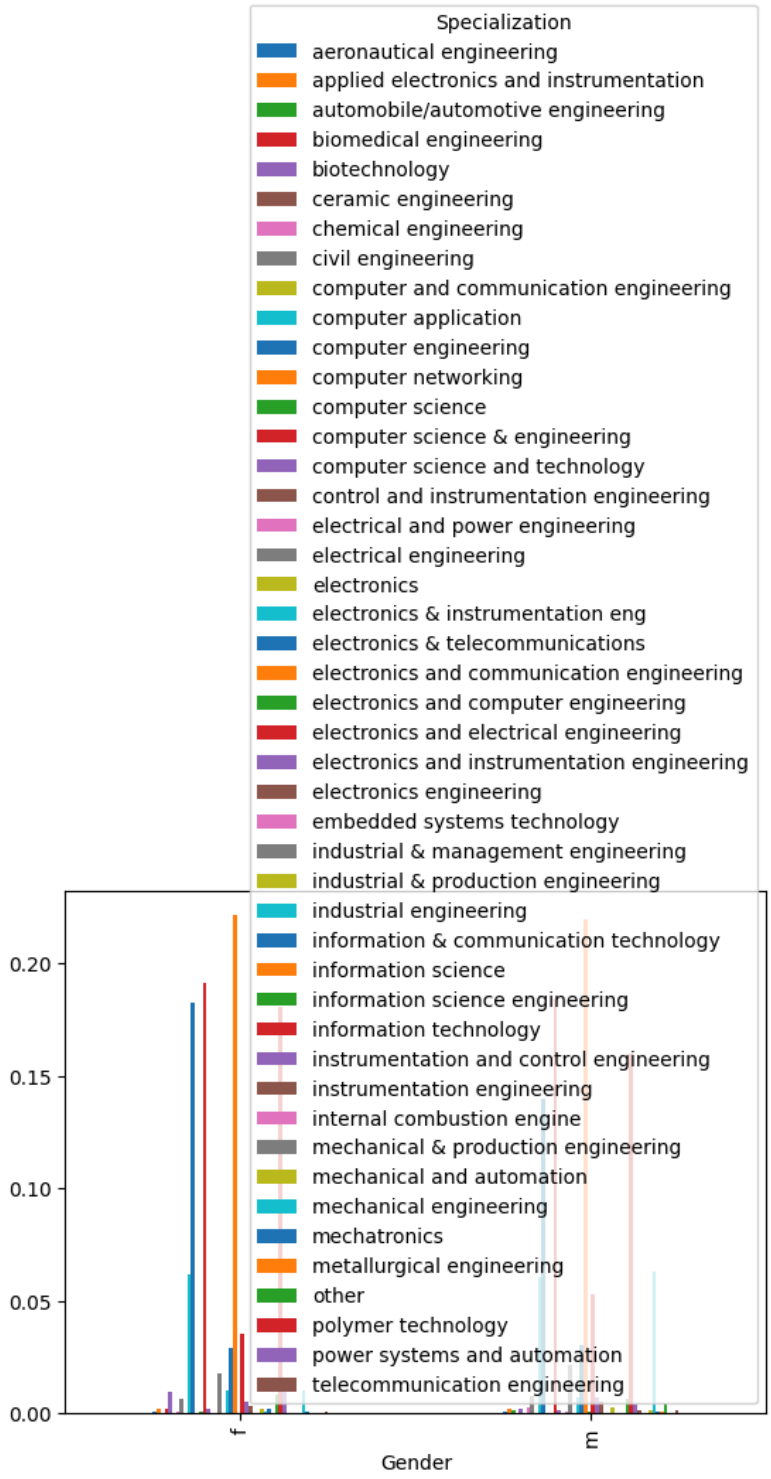
```
pd.crosstab(df["Degree"],df["Designation"])
```

Designation	.net developer	.net web developer	account executive	account manager	admin assistant	administrative coordinator	administrative support	aircraft technician	and devel
Degree									
B.Tech/B.E.	29	3	4	1	2	1	0	1	
M.Sc. (Tech.)	0	0	0	0	0	0	0	0	
M.Tech./M.E.	1	0	0	0	0	0	0	0	
MCA	4	1	0	0	0	0	1	0	

4 rows × 419 columns

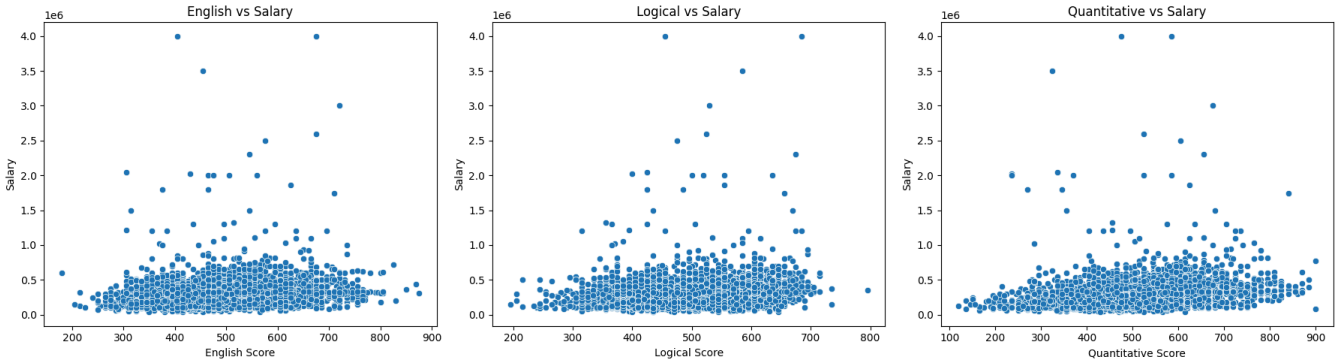
```
tab = pd.crosstab(df['Gender'], df['Specialization'], normalize='index')

tab.plot(kind='bar')
```

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




```
plt.figure(figsize=(18, 5))

# English vs Salary
plt.subplot(1, 3, 1)
sns.scatterplot(data=df, x='English', y='Salary')
plt.title('English vs Salary')
plt.xlabel('English Score')
plt.ylabel('Salary')

# Logical vs Salary
plt.subplot(1, 3, 2)
sns.scatterplot(data=df, x='Logical', y='Salary')
plt.title('Logical vs Salary')
plt.xlabel('Logical Score')
plt.ylabel('Salary')

# Quantitative vs Salary
plt.subplot(1, 3, 3)
sns.scatterplot(data=df, x='Quant', y='Salary')
```

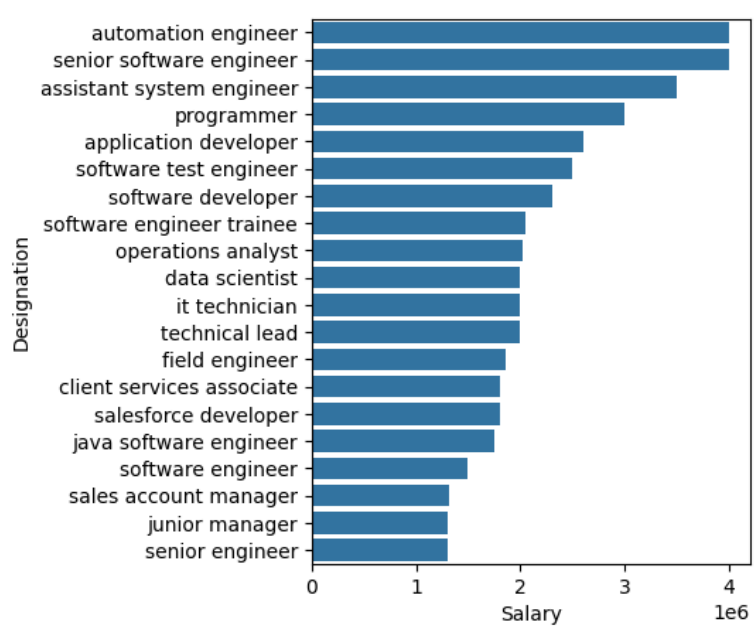
```
plt.title('Quantitative vs Salary')
plt.xlabel('Quantitative Score')
plt.ylabel('Salary')

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



Which top 50 jobs Designation has more salary in IT companies ?

```
df_destignation = df.groupby('Designation')['Salary'].max().sort_values(ascending=False).reset_index().head(20)
```

```
plt.figure(figsize=(4,5))
sns.barplot(y='Designation', x='Salary', data=df_destignation)
plt.show()
```




```
group = df.groupby('Gender')
```

```
group['Salary'].agg(['min', 'max', 'mean', 'median'])
```

Gender	min	max	mean	median
f	35000.0	3500000.0	294937.304075	300000.0
m	35000.0	4000000.0	311716.211772	300000.0

```
df.boxplot(by='Gender', column='Salary')
```