AMCAT Exploratory Data Analysis (EDA)

Data description

- The dataset was released by Aspiring Minds from the Aspiring Mind Employment Outcome 2015 (AMEO). The study is primarily limited only to students with engineering disciplines. 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. The dataset contains around 40 independent variables and 4000 data points. The independent variables are both continuous and categorical in nature. The dataset contains a unique identifier for each candidate. Below mentioned table contains the details for the original dataset.

Summary Table for dataset

VARIABLES	TYPE	Description
ID	UID	A unique ID to identify a candidate
Salary	Continuous	Annual CTC offered to the candidate (in INR)
DOJ	Date	Date of joining the company
DOL	Date	Date of leaving the company
Designation	Categorical	Designation offered in the job
JobCity	Categorical	Location of the job (city)
Gender	Categorical	Candidate’s gender
DOB	Date	Date of birth of candidate
10percentage	Continuous	Overall marks obtained in grade 10 examinations
10board	Continuous	The school board whose curriculum the candidate followed in grade 10
12graduation	Date	Year of graduation - senior year high school
12percentage	Continuous	Overall marks obtained in grade 12 examinations
12board	Date	The school board whose curriculum the candidate followed in grade 12
CollegeID	NA/ID	Unique ID identifying the college which the candidate at tended
CollegeTier	Categorical	Tier of college
Degree	Categorical	Degree obtained/pursued by the candidate
Specialization	Categorical	Specialization pursued by the candidate
CollegeGPA	Continuous	Aggregate GPA at graduation
CollegeCityID	NA/ID	A unique ID to identify the ci in which the c oll ege is located
CollegeCityTier	Categorical	The tier of the city in which the college is locateded

VARIABLES	TYPE	Description
CollegeState	Categorical	Name of States
GraduationYear	Date	Year of graduation (Bachelor's degree)
English	Continuous	Scores in AMCAT English section
Logical	Continuous	Scores in AMCAT Logical section
Quant	Continuous	Scores in AMCAT Quantitative section
Domain	Continuous/ Standardized	Scores in AMCAT's domain module
ComputerProgramming	Continuous	Score in AMCAT' Computer programming secti on
ElectronicsAndSemicon	Continuous	Score in AMC's Electronics & Semicondu c tor Engineering section
ComputerScience	Continuous	Score in AMCAT's Computer Science section
MechanicalEngg	Continuous	Score in AMCAT's Mechanical Engineering section
ElectricalEngg	Continuous	Score in AMCAT's Electrical Engineering section
TelecomEngg	Continuous	Score in AMCAT's Telecommunication Eng ineering section
CivilEngg	Continuous	Scoe in AMCAT's Civil Engineering se ction
conscientiousness	Continuous/ Standardized	Scores in one of the sections of CAT's personality test
agreeableness	Continuous/ Standardize d	Scores in one of the sections ofAMCAT's personality test
extraversion	Continuous/ Standardifzed	Scores in one of the sections o AMCAT's personality test
neuroticism	Continuous/ Standarodized	Scores in one of the sections f AMCAT's personality test
openness_to_experience	Contthe sections of AMCAT's personality testan rd ized	Scores in one of th

Objective

1. Understand Dataset Structure: Explore the overall structure of the AMCAT dataset, including the types and distribution of variables related to candidates' profiles.

2. Gender and Specialization Analysis: Analyze the relationship between gender and specialization to uncover any trends or preferences in candidate specialization choices.

3. Salary as Target Variable: Investigate the factors influencing salary, focusing on identifying trends, correlations, and patterns between independent variables (such as specialization, gender, etc.) and the target variable (Salary).
4. Feature Distribution and Outliers: Visualize and interpret the distribution of various features (e.g., experience, skills, etc.), identifying outliers or unusual patterns in the dataset.
5. Insights and Correlations: Summarize key insights and relationships, particularly looking for dependencies between variables like gender, specialization, and their impact on salary.

Importing required libraries and loading data for the project

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import os
import datetime as dt
import warnings
warnings.filterwarnings('ignore')

print (os.getcwd())
```

```
C:\Users\parmo\DS_internship\Project 1
```

```
# Loading Data
df=pd.read_csv("data.csv")
```

Data overview and summary statistics Use the following methods and attributes on thedataframe:

1. head()
2. shape
3. columns
4. info()
5. describe()

```
display(df.head())
```

```
  Unnamed: 0      ID     Salary          DOJ          DOL  \
0      train  203097   420000.0  6/1/12 0:00      present
1      train  579905   500000.0  9/1/13 0:00      present
2      train  810601   325000.0  6/1/14 0:00      present
```

```
3      train  267447  1100000.0  7/1/11 0:00      present
4      train  343523   200000.0  3/1/14 0:00  3/1/15 0:00

                 Designation    JobCity Gender           DOB
10percentage  \
0   senior quality engineer  Bangalore      f  2/19/90 0:00
84.3
1         assistant manager     Indore      m  10/4/89 0:00
85.4
2          systems engineer    Chennai      f   8/3/92 0:00
85.0
3  senior software engineer    Gurgaon      m  12/5/89 0:00
85.6
4                       get    Manesar      m  2/27/91 0:00
78.0

   ... ComputerScience  MechanicalEngg  ElectricalEngg TelecomEngg
CivilEngg  \
0  ...              -1              -1              -1          -1
-1
1  ...              -1              -1              -1          -1
-1
2  ...              -1              -1              -1          -1
-1
3  ...              -1              -1              -1          -1
-1
4  ...              -1              -1              -1          -1
-1

   conscientiousness agreeableness extraversion  nueroticism  \
0             0.9737        0.8128       0.5269      1.35490
1            -0.7335        0.3789       1.2396     -0.10760
2             0.2718        1.7109       0.1637     -0.86820
3             0.0464        0.3448      -0.3440     -0.40780
4            -0.8810       -0.2793      -1.0697      0.09163

   openess_to_experience
0                -0.4455
1                 0.8637
2                 0.6721
3                -0.9194
4                -0.1295

[5 rows x 39 columns]

display (df.shape)  # Number of records- Shape

(3998, 39)

display (df.columns)  # Display the 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')
```

```
print (df.info()) # Data set details – 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
None

display (df.describe()) # Data set details – Describe

                 ID        Salary  10percentage  12graduation
12percentage  \
count  3.998000e+03  3.998000e+03   3998.000000   3998.000000
3998.000000
mean   6.637945e+05  3.076998e+05     77.925443   2008.087544
74.466366
std    3.632182e+05  2.127375e+05      9.850162      1.653599
10.999933
min    1.124400e+04  3.500000e+04     43.000000   1995.000000
40.000000
25%    3.342842e+05  1.800000e+05     71.680000   2007.000000
66.000000
50%    6.396000e+05  3.000000e+05     79.150000   2008.000000
74.400000
75%    9.904800e+05  3.700000e+05     85.670000   2009.000000
82.600000
max    1.298275e+06  4.000000e+06     97.760000   2013.000000
98.700000

          CollegeID  CollegeTier   collegeGPA  CollegeCityID
CollegeCityTier  \
count   3998.000000  3998.000000  3998.000000    3998.000000
3998.000000
mean    5156.851426     1.925713    71.486171    5156.851426
0.300400
std     4802.261482     0.262270     8.167338    4802.261482
0.458489
min        2.000000     1.000000     6.450000       2.000000
0.000000
25%      494.000000     2.000000    66.407500     494.000000
0.000000
```

```
50%     3879.000000     2.000000    71.720000    3879.000000
0.000000
75%     8818.000000     2.000000    76.327500    8818.000000
1.000000
max    18409.000000     2.000000    99.930000   18409.000000
1.000000

       ...  ComputerScience  MechanicalEngg  ElectricalEngg
TelecomEngg  \
count  ...      3998.000000     3998.000000     3998.000000
3998.000000
mean   ...        90.742371       22.974737       16.478739
31.851176
std    ...       175.273083       98.123311       87.585634
104.852845
min    ...        -1.000000       -1.000000       -1.000000    -
1.000000
25%    ...        -1.000000       -1.000000       -1.000000    -
1.000000
50%    ...        -1.000000       -1.000000       -1.000000    -
1.000000
75%    ...        -1.000000       -1.000000       -1.000000    -
1.000000
max    ...       715.000000      623.000000      676.000000
548.000000

         CivilEngg  conscientiousness  agreeableness  extraversion  \
count  3998.000000        3998.000000    3998.000000   3998.000000
mean      2.683842          -0.037831       0.146496      0.002763
std      36.658505           1.028666       0.941782      0.951471
min      -1.000000          -4.126700      -5.781600     -4.600900
25%      -1.000000          -0.713525      -0.287100     -0.604800
50%      -1.000000           0.046400       0.212400      0.091400
75%      -1.000000           0.702700       0.812800      0.672000
max     516.000000           1.995300       1.904800      2.535400

       nueroticism  openess_to_experience
count  3998.000000            3998.000000
mean     -0.169033              -0.138110
std       1.007580               1.008075
min      -2.643000              -7.375700
25%      -0.868200              -0.669200
50%      -0.234400              -0.094300
75%       0.526200               0.502400
max       3.352500               1.822400

[8 rows x 27 columns]
```

```
display (df.isna().sum() ) #Checking null values
```

```
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
```

Timestamp format is used for the columns DOJ, DOL, and DOB.

```
df["DOJ"]=pd.to_datetime(df["DOJ"]).dt.date
df["DOL"].replace("present",dt.datetime.today(),inplace=True)
df['DOL'] = pd.to_datetime(df['DOL']).dt.date
## We will engineer this feature from DOJ and DOL as we are only
concerned with␣how many years the person has worked
## in the organization.
```

```
df['Experience'] = pd.to_datetime(df["DOL"]).dt.year -
pd.to_datetime(df['DOJ']).dt.year
##We only need DOB year,so we will convert DOB column from timestamp
to year
df['DOB'] = pd.to_datetime(df['DOB']).dt.year
df1=(df.head(5))

df.dtypes # data types of each column
```

```
Unnamed: 0                 object
ID                          int64
Salary                    float64
DOJ                        object
DOL                        object
Designation                object
JobCity                    object
Gender                     object
DOB                         int32
10percentage              float64
10board                    object
12graduation                int64
12percentage              float64
12board                    object
CollegeID                   int64
CollegeTier                 int64
Degree                     object
Specialization             object
collegeGPA                float64
CollegeCityID               int64
CollegeCityTier             int64
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
Experience                  int32
dtype: object
```

Univariate Analysis -> PDF, Histograms, Boxplots, Countplots, etc..

1. Find the outliers in each numerical column
2. Understand the probability and frequency distribution of each numerical column
3. Understand the frequency distribution of each categorical Variable/Column
4. Mention observations after each plot. = print(" ")
5. Cat Col = Num Col -> Analyzing relationships between categorical and numerical variables.
6. Cat=Num, Num=Num, Cat=Cat

```
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_df = df.select_dtypes(include=['object'])
discrete_univariate_analysis(discrete_df)
print("Discrete Univariate analysis provides insigts the cat.
variables , reverlind the count of unique values & frequency
distribution ")
```

```
********** Unnamed: 0 **********
count         3998
nunique          1
unique     [train]
Name: Unnamed: 0, dtype: object
Value Counts:
 Unnamed: 0
train    3998
Name: count, dtype: int64

********** DOJ **********
count                                                  3998
nunique                                                  81
unique     [2012-06-01, 2013-09-01, 2014-06-01, 2011-07-0...
Name: DOJ, dtype: object
Value Counts:
 DOJ
2014-07-01    199
2014-06-01    180
2014-08-01    178
2014-09-01    142
2014-01-01    142
             ...
2015-11-01      1
2009-11-01      1
2004-08-01      1
```

```
2009-09-01      1
2007-02-01      1
Name: count, Length: 81, dtype: int64

********** DOL **********
count                                                    3998
nunique                                                    67
unique     [2024-10-03, 2015-03-01, 2015-05-01, 2015-07-0...
Name: DOL, dtype: object
Value Counts:
 DOL
2024-10-03    1875
2015-04-01     573
2015-03-01     124
2015-05-01     112
2015-01-01      99
              ...
2005-03-01       1
2015-10-01       1
2010-02-01       1
2011-02-01       1
2010-10-01       1
Name: count, Length: 67, dtype: int64

********** Designation **********
count                                                    3998
nunique                                                   419
unique     [senior quality engineer, assistant manager, s...
Name: Designation, dtype: object
Value Counts:
 Designation
software engineer                       539
software developer                      265
system engineer                         205
programmer analyst                      139
systems engineer                        118
                                       ...
cad drafter                               1
noc engineer                              1
human resources intern                    1
senior quality assurance engineer         1
jr. software developer                    1
Name: count, Length: 419, dtype: int64

********** JobCity **********
count                                                    3998
nunique                                                   339
unique     [Bangalore, Indore, Chennai, Gurgaon, Manesar,...
Name: JobCity, dtype: object
Value Counts:
```

```
 JobCity
Bangalore           627
-1                  461
Noida               368
Hyderabad           335
Pune                290
                   ...
Tirunelvelli          1
Ernakulam             1
Nanded                1
Dharmapuri            1
Asifabadbanglore      1
Name: count, Length: 339, dtype: int64

********** Gender **********
count        3998
nunique         2
unique     [f, m]
Name: Gender, dtype: object
Value Counts:
 Gender
m    3041
f     957
Name: count, dtype: int64

********** 10board **********
count                                                    3998
nunique                                                   275
unique     [board ofsecondary education,ap, cbse, state b...
Name: 10board, dtype: object
Value Counts:
 10board
cbse                           1395
state board                    1164
0                               350
icse                            281
ssc                             122
                               ...
hse,orissa                        1
national public school            1
nagpur board                      1
jharkhand academic council        1
bse,odisha                        1
Name: count, Length: 275, dtype: int64

********** 12board **********
count                                                    3998
nunique                                                   340
unique     [board of intermediate education,ap, cbse, sta...
Name: 12board, dtype: object
```

```
Value Counts:
 12board
cbse                                  1400
state board                           1254
0                                      359
icse                                   129
up board                                87
                                       ...
jawahar higher secondary school          1
nagpur board                             1
bsemp                                    1
board of higher secondary orissa         1
boardofintermediate                      1
Name: count, Length: 340, dtype: int64

********** Degree **********
count                                                 3998
nunique                                                  4
unique     [B.Tech/B.E., MCA, M.Tech./M.E., M.Sc. (Tech.)]
Name: Degree, dtype: object
Value Counts:
 Degree
B.Tech/B.E.      3700
MCA               243
M.Tech./M.E.       53
M.Sc. (Tech.)       2
Name: count, dtype: int64

********** Specialization **********
count                                                    3998
nunique                                                    46
unique     [computer engineering, electronics and communi...
Name: Specialization, dtype: object
Value Counts:
 Specialization
electronics and communication engineering      880
computer science & engineering                 744
information technology                         660
computer engineering                           600
computer application                           244
mechanical engineering                         201
electronics and electrical engineering         196
electronics & telecommunications               121
electrical engineering                          82
electronics & instrumentation eng               32
civil engineering                               29
electronics and instrumentation engineering     27
information science engineering                 27
instrumentation and control engineering         20
```

```
electronics engineering                          19
biotechnology                                    15
other                                            13
industrial & production engineering              10
applied electronics and instrumentation           9
chemical engineering                              9
computer science and technology                   6
telecommunication engineering                     6
mechanical and automation                         5
automobile/automotive engineering                 5
instrumentation engineering                       4
mechatronics                                      4
aeronautical engineering                          3
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
nunique                                                26
unique     [Andhra Pradesh, Madhya Pradesh, Uttar Pradesh...
Name: CollegeState, dtype: object
Value Counts:
 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
```

```
Discrete Univariate analysis provides insigts the cat. variables ,
reverlind the count of unique values & frequency distribution
```

```
def numerical_univariate_analysis(numerical_data):
    for col_name in numerical_data:
        print("*"*10, col_name, "*"*10)
        print(numerical_data[col_name].agg(['count', 'min', 'max',
'mean', 'median', 'std', 'skew']))
        print()
numerical_df = df.select_dtypes(include=['float64', 'int64','int32'])
numerical_univariate_analysis(numerical_df)
print("Numerical univeriate analysis summary of statistics key (count,
min, max, mean,'median', std, skew ) for num columns indicating the
central tendency , variablility and skewness of distribution to helps
identify the potentail outliers& data data distribution shapes")
```

```
********** ID **********
count     3.998000e+03
min       1.124400e+04
max       1.298275e+06
mean      6.637945e+05
median    6.396000e+05
std       3.632182e+05
skew      5.477047e-02
Name: ID, dtype: float64

********** Salary **********
count     3.998000e+03
min       3.500000e+04
max       4.000000e+06
```

```
mean      3.076998e+05
median    3.000000e+05
std       2.127375e+05
skew      6.451081e+00
Name: Salary, dtype: float64

********** DOB **********
count     3998.000000
min       1977.000000
max       1997.000000
mean      1990.427464
median    1991.000000
std          1.767473
skew        -0.887271
Name: DOB, dtype: float64

********** 10percentage **********
count     3998.000000
min         43.000000
max         97.760000
mean        77.925443
median      79.150000
std          9.850162
skew        -0.591019
Name: 10percentage, dtype: float64

********** 12graduation **********
count     3998.000000
min       1995.000000
max       2013.000000
mean      2008.087544
median    2008.000000
std          1.653599
skew        -0.964090
Name: 12graduation, dtype: float64

********** 12percentage **********
count     3998.000000
min         40.000000
max         98.700000
mean        74.466366
median      74.400000
std         10.999933
skew        -0.032607
Name: 12percentage, dtype: float64

********** CollegeID **********
count      3998.000000
min           2.000000
max       18409.000000
```

```
mean       5156.851426
median     3879.000000
std        4802.261482
skew          0.649176
Name: CollegeID, dtype: float64

********** CollegeTier **********
count     3998.000000
min          1.000000
max          2.000000
mean         1.925713
median       2.000000
std          0.262270
skew        -3.247991
Name: CollegeTier, dtype: float64

********** collegeGPA **********
count     3998.000000
min          6.450000
max         99.930000
mean        71.486171
median      71.720000
std          8.167338
skew        -1.249209
Name: collegeGPA, dtype: float64

********** CollegeCityID **********
count      3998.000000
min           2.000000
max       18409.000000
mean       5156.851426
median     3879.000000
std        4802.261482
skew          0.649176
Name: CollegeCityID, dtype: float64

********** CollegeCityTier **********
count     3998.000000
min          0.000000
max          1.000000
mean         0.300400
median       0.000000
std          0.458489
skew         0.871120
Name: CollegeCityTier, dtype: float64

********** GraduationYear **********
count     3998.000000
min          0.000000
max       2017.000000
```

```
mean      2012.105803
median    2013.000000
std         31.857271
skew       -63.068064
Name: GraduationYear, dtype: float64

********** English **********
count     3998.000000
min        180.000000
max        875.000000
mean       501.649075
median     500.000000
std        104.940021
skew         0.191997
Name: English, dtype: float64

********** Logical **********
count     3998.000000
min        195.000000
max        795.000000
mean       501.598799
median     505.000000
std         86.783297
skew        -0.216602
Name: Logical, dtype: float64

********** Quant **********
count     3998.000000
min        120.000000
max        900.000000
mean       513.378189
median     515.000000
std        122.302332
skew        -0.019399
Name: Quant, dtype: float64

********** Domain **********
count     3998.000000
min         -1.000000
max          0.999910
mean         0.510490
median       0.622643
std          0.468671
skew        -1.922146
Name: Domain, dtype: float64

********** ComputerProgramming **********
count     3998.000000
min         -1.000000
max        840.000000
```

```
mean       353.102801
median     415.000000
std        205.355519
skew        -0.778106
Name: ComputerProgramming, dtype: float64

********** ElectronicsAndSemicon **********
count     3998.000000
min         -1.000000
max        612.000000
mean        95.328414
median      -1.000000
std        158.241218
skew         1.195975
Name: ElectronicsAndSemicon, dtype: float64

********** ComputerScience **********
count     3998.000000
min         -1.000000
max        715.000000
mean        90.742371
median      -1.000000
std        175.273083
skew         1.529521
Name: ComputerScience, dtype: float64

********** MechanicalEngg **********
count     3998.000000
min         -1.000000
max        623.000000
mean        22.974737
median      -1.000000
std         98.123311
skew         4.029563
Name: MechanicalEngg, dtype: float64

********** ElectricalEngg **********
count     3998.000000
min         -1.000000
max        676.000000
mean        16.478739
median      -1.000000
std         87.585634
skew         5.060407
Name: ElectricalEngg, dtype: float64

********** TelecomEngg **********
count     3998.000000
min         -1.000000
max        548.000000
```

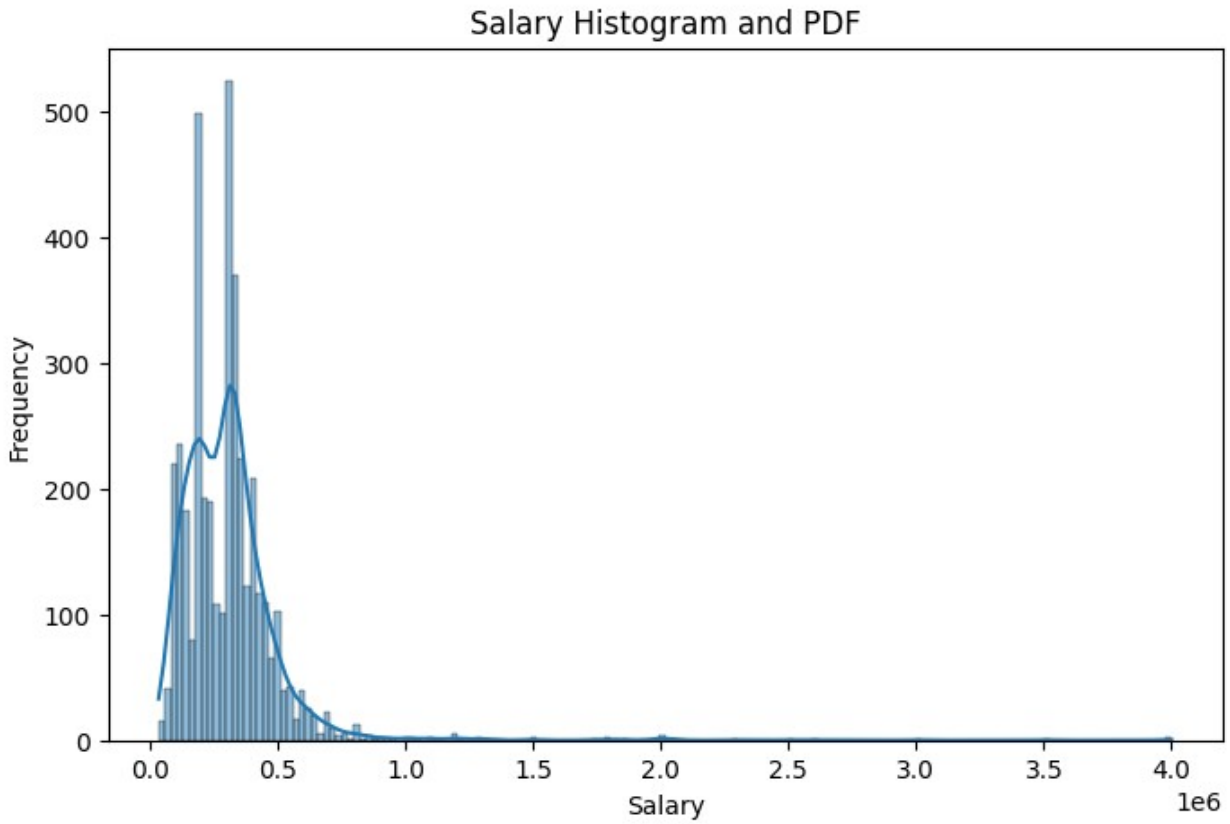
```
mean        31.851176
median      -1.000000
std        104.852845
skew         3.041261
Name: TelecomEngg, dtype: float64

********** CivilEngg **********
count     3998.000000
min         -1.000000
max        516.000000
mean         2.683842
median      -1.000000
std         36.658505
skew        10.315681
Name: CivilEngg, dtype: float64

********** conscientiousness **********
count     3998.000000
min         -4.126700
max          1.995300
mean        -0.037831
median       0.046400
std          1.028666
skew        -0.527003
Name: conscientiousness, dtype: float64

********** agreeableness **********
count     3998.000000
min         -5.781600
max          1.904800
mean         0.146496
median       0.212400
std          0.941782
skew        -1.204915
Name: agreeableness, dtype: float64

********** extraversion **********
count     3998.000000
min         -4.600900
max          2.535400
mean         0.002763
median       0.091400
std          0.951471
skew        -0.523267
Name: extraversion, dtype: float64

********** nueroticism **********
count     3998.000000
min         -2.643000
max          3.352500
```

```
mean         -0.169033
median       -0.234400
std           1.007580
skew          0.165710
Name: nueroticism, dtype: float64

********** openess_to_experience **********
count     3998.000000
min         -7.375700
max          1.822400
mean        -0.138110
median      -0.094300
std          1.008075
skew        -1.506962
Name: openess_to_experience, dtype: float64

********** Experience **********
count     3998.000000
min          0.000000
max         33.000000
mean         5.755128
median       3.000000
std          4.789783
skew         0.188965
Name: Experience, dtype: float64

Numerical univeriate analysis summary of statistics key (count, min,
max, mean,'median', std, skew ) for num columns indicating the central
tendency , variablility and skewness of distribution to helps identify
the potentail outliers& data data distribution shapes

cat_col = []  # # Identifying categorical columns
for x in df.dtypes.index:
    if df.dtypes[x] == 'object':
        cat_col.append(x)
display(cat_col)

['Unnamed: 0',
 'DOJ',
 'DOL',
 'Designation',
 'JobCity',
 'Gender',
 '10board',
 '12board',
 'Degree',
 'Specialization',
 'CollegeState']
```

```
# Select only the numerical columns
num_columns = df.select_dtypes(include=['float64','int64','int32'])

def find_outliers(df):
    outliers_dict = {}
    for column in df.columns:
        Q1 = df[column].quantile(0.25)
        Q3 = df[column].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        outliers = df[(df[column] < lower_bound) | (df[column] >
upper_bound)]
        outliers_dict[column] = outliers.shape[0]  # Number of
outliers
    return outliers_dict

outliers_info = find_outliers(num_columns)
print(outliers_info)
```

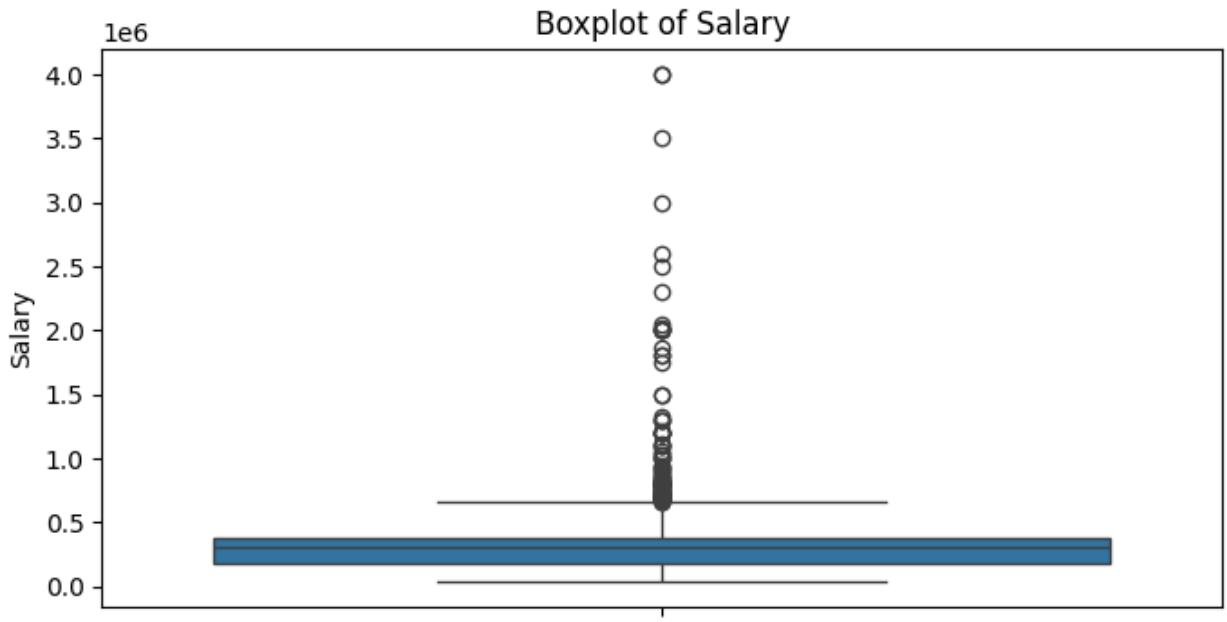
```
{'ID': 0, 'Salary': 109, 'DOB': 22, '10percentage': 30,
'12graduation': 45, '12percentage': 1, 'CollegeID': 0, 'CollegeTier':
297, 'collegeGPA': 38, 'CollegeCityID': 0, 'CollegeCityTier': 0,
'GraduationYear': 2, 'English': 15, 'Logical': 18, 'Quant': 25,
'Domain': 246, 'ComputerProgramming': 2, 'ElectronicsAndSemicon': 2,
'ComputerScience': 902, 'MechanicalEngg': 235, 'ElectricalEngg': 161,
'TelecomEngg': 374, 'CivilEngg': 42, 'conscientiousness': 39,
'agreeableness': 123, 'extraversion': 40, 'nueroticism': 15,
'openess_to_experience': 95, 'Experience': 1}
```

```
# Plotting the Salary Histogram and PDF
plt.figure(figsize=(8, 5))
sns.histplot(df['Salary'], kde=True)  # Histogram + Kernel Density
Estimation (KDE) for PDF
plt.title('Salary Histogram and PDF')
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.show()
print("Histogram of salaries shows right skewed distribution most
candidates earn lower salaries & other earning higher salaries.")
```



Histogram of salaries shows right skewed distribution most candidates earn lower salaries & other earning higher salaries.

```
# Plotting the Salary Boxplot
plt.figure(figsize=(8, 4))
sns.boxplot(df['Salary'])
plt.title('Boxplot of Salary')
plt.show()
print("boxplot of Salary reveals a right-skewed distribution with median lower than maximum indicating the presence of high earners that elevate overall range of salaries")
```




```
boxplot of Salary reveals a right-skewed distribution with median
lower than maximum indicating the presence of high earners that
elevate overall range of salaries
```

```
# Detecting outliers using IQR for Salary
Q1 = df['Salary'].quantile(0.25)
Q3 = df['Salary'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df['Salary'] < lower_bound) | (df['Salary'] >
upper_bound)]
print(f"Number of outliers in Salary: {len(outliers)}")
```

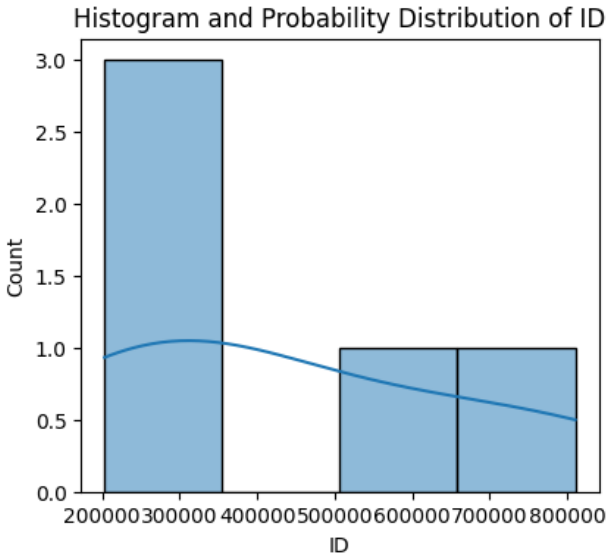
```
Number of outliers in Salary: 109
```

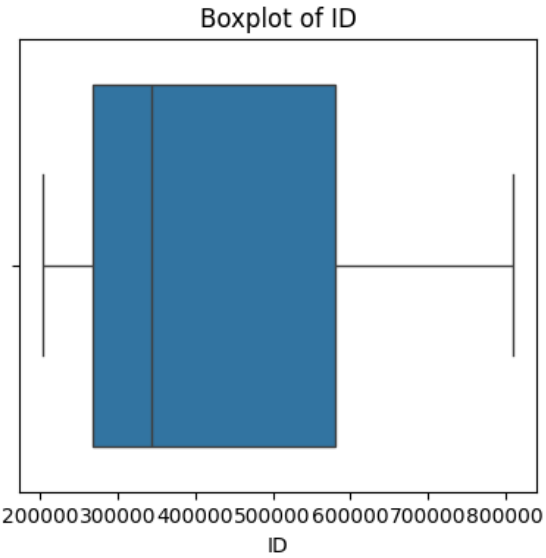
Understand the probability and frequency distribution of each numerical columns

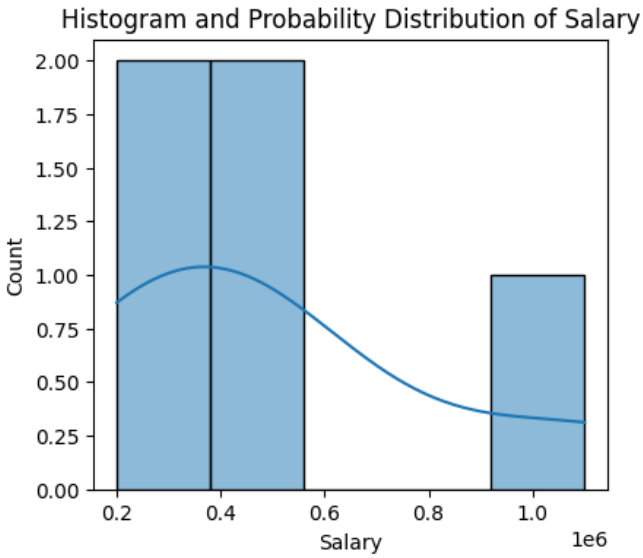
- Probability distribution of num columns can reveal the data follows a normal distribution with skewness indicating portential outliers or asymmetry (statistical modeling & inference).
- Frequency Distribution provides insights how each value or range of values occurs allowing for identification of patterns, treands, potential clusters within the data (understanding relationships with target variables like salary).

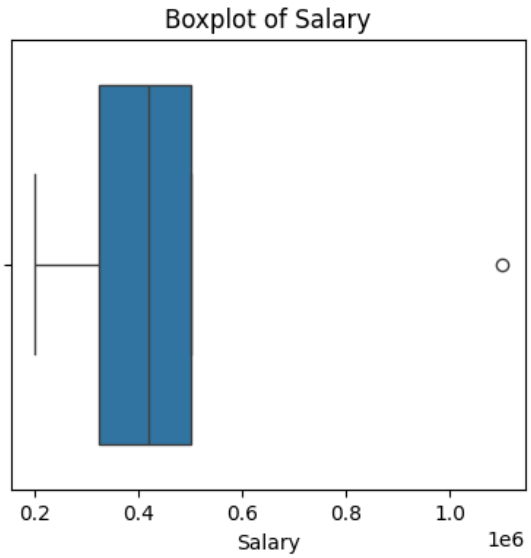
```
# List of numerical columns
num_columns = df1.select_dtypes(include=['float64',
'int64','int32']).columns
```

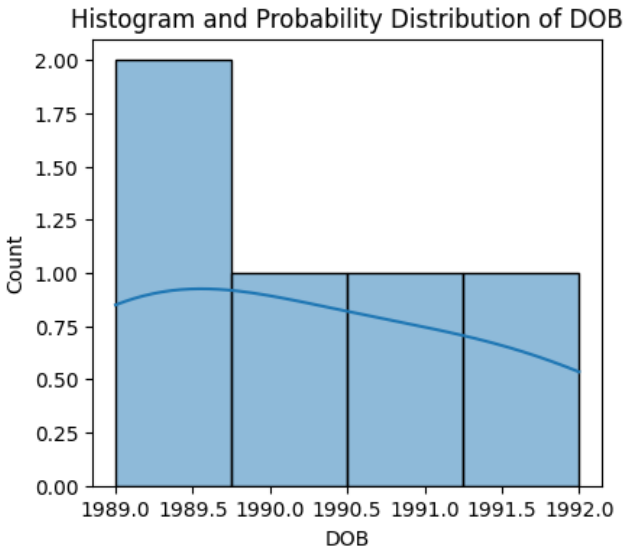
```python
# Plotting each numerical column
for column in num_columns:
    plt.figure(figsize=(10, 4))

    # Histogram and KDE (PDF)
    plt.subplot(1, 2, 1)
    sns.histplot(df1[column], kde=True)
    plt.title(f'Histogram and Probability Distribution of {column}')

    # Boxplot for detecting outliers
    plt.subplot(1, 2, 2)
    sns.boxplot(x=df1[column])
    plt.title(f'Boxplot of {column}')
    plt.show()
```

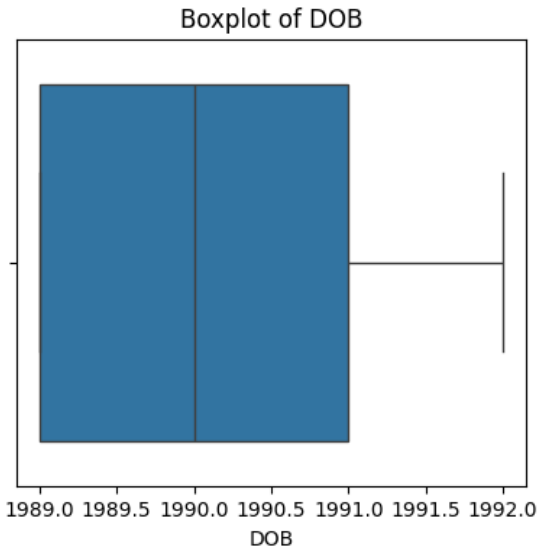


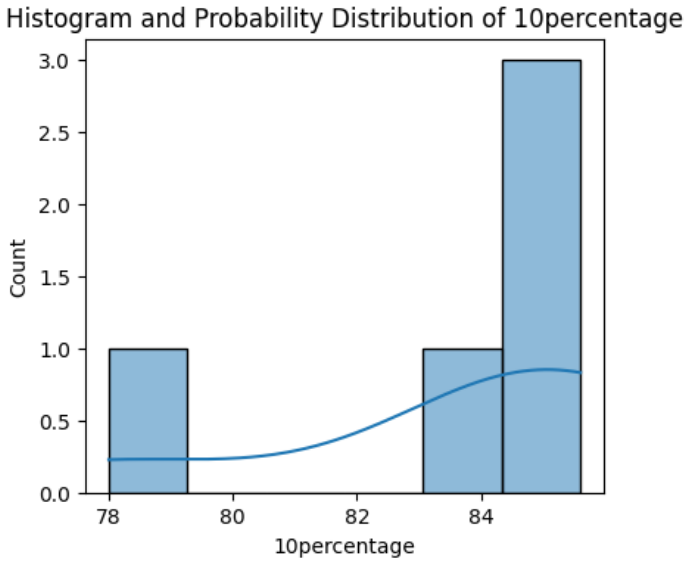


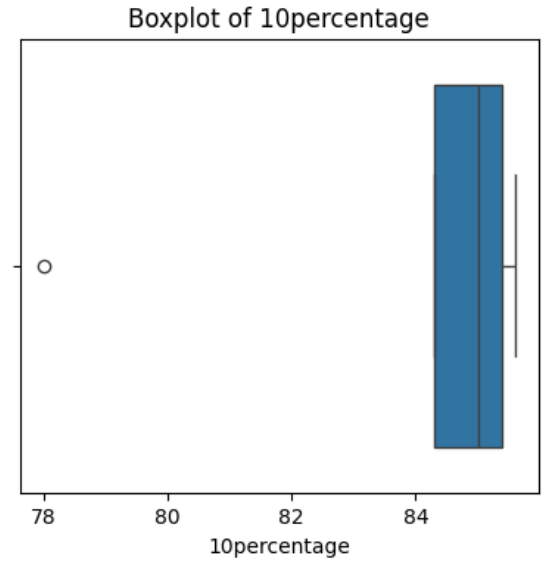


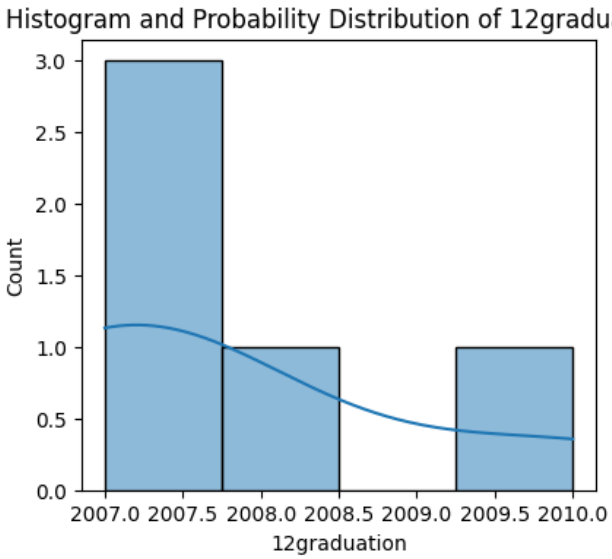


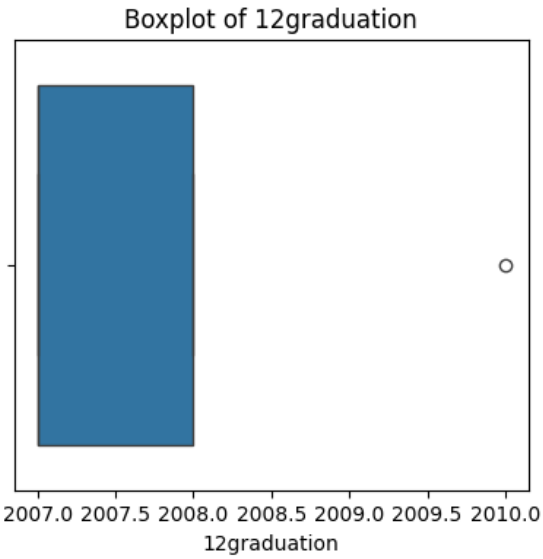

Histogram and Probability Distribution of DOB
Count
DOB
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1989.0 1989.5 1990.0 1990.5 1991.0 1991.5 1992.0

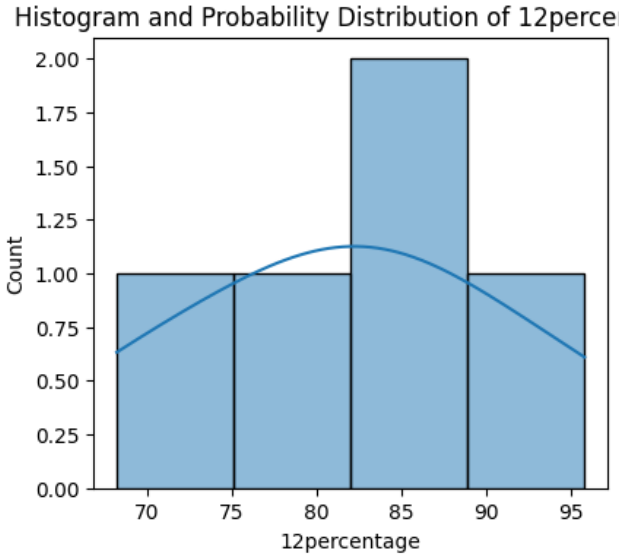

Boxplot of DOB
DOB
1989.0 1989.5 1990.0 1990.5 1991.0 1991.5 1992.0

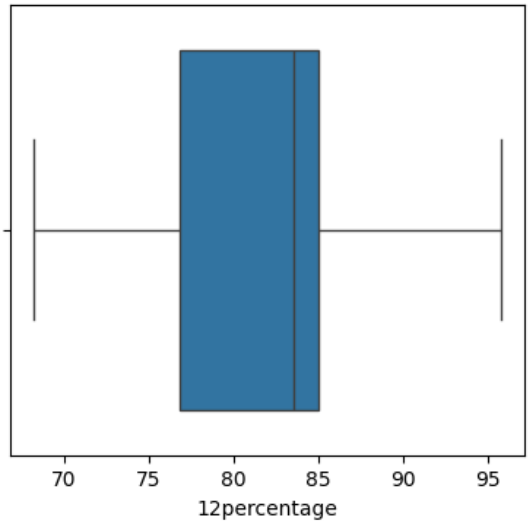

Histogram and Probability Distribution of 10percentage
Count
10percentage
3.0
2.5
2.0
1.5
1.0
0.5
0.0
78
80
82
84

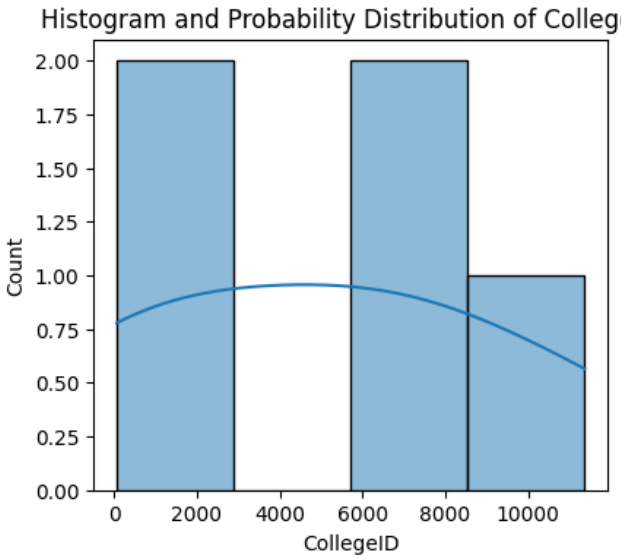

Boxplot of 10percentage
10percentage
78
80
82
84

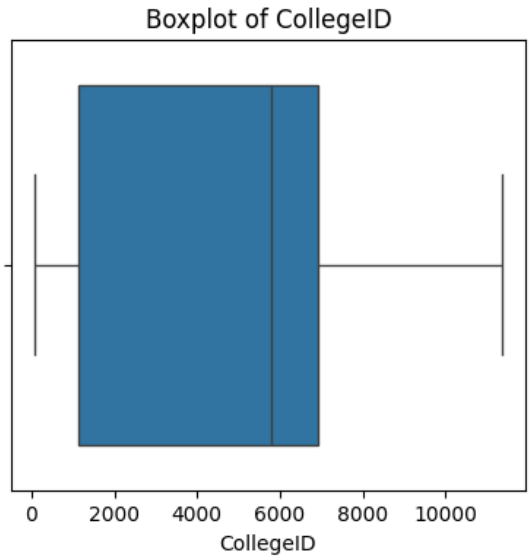

Histogram and Probability Distribution of 12graduation
Count
12graduation
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2007.0 2007.5 2008.0 2008.5 2009.0 2009.5 2010.0

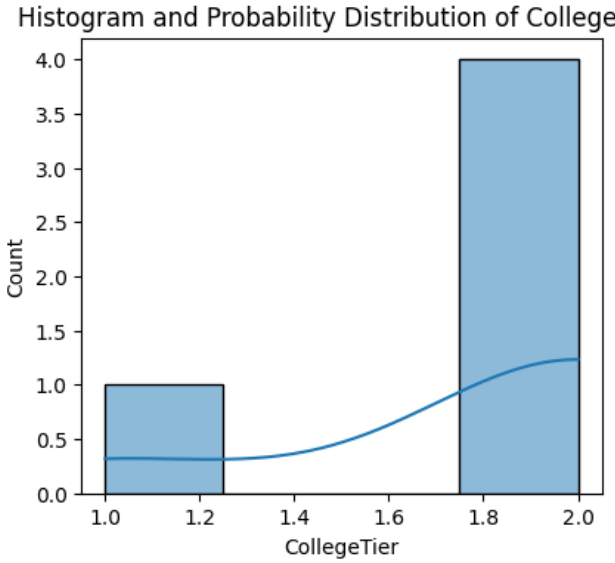

Boxplot of 12graduation
12graduation
2007.0 2007.5 2008.0 2008.5 2009.0 2009.5 2010.0

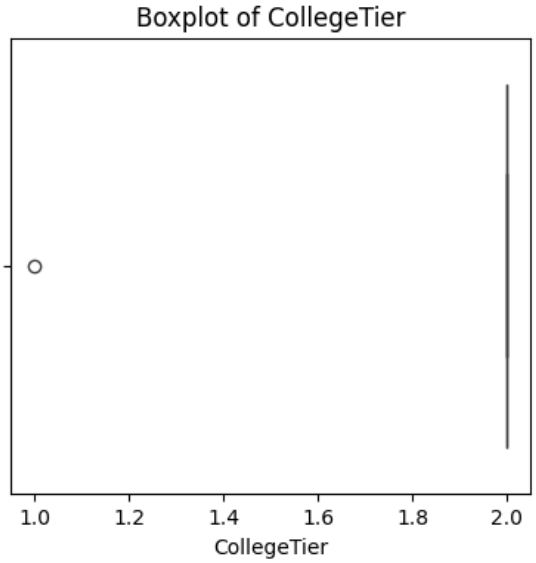

Histogram and Probability Distribution of 12percentage
Count
12percentage
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
70 75 80 85 90 95

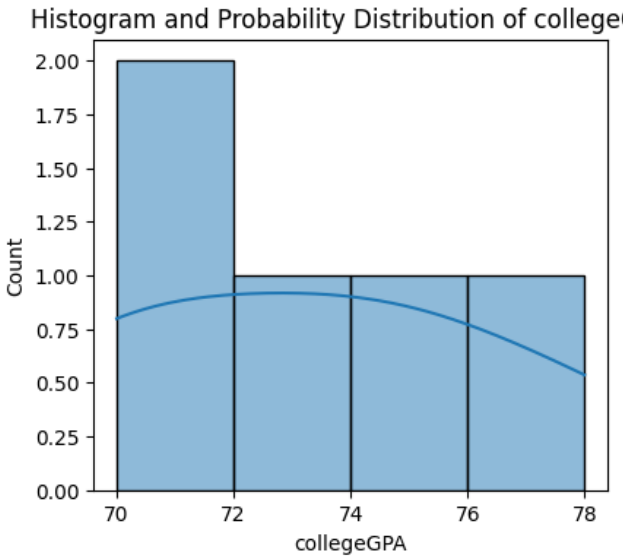

Boxplot of 12percentage
12percentage
70 75 80 85 90 95

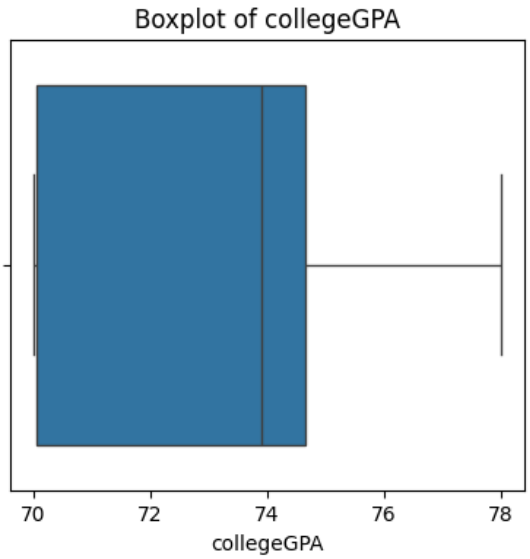

Histogram and Probability Distribution of CollegeID
Count
CollegeID
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0
2000
4000
6000
8000
10000

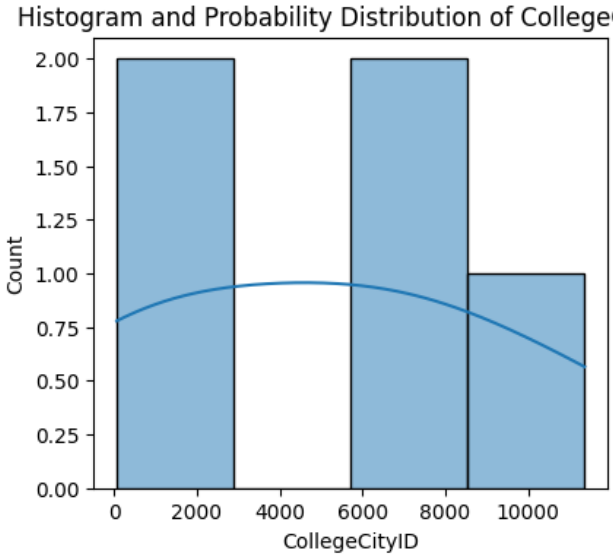

Boxplot of CollegeID
CollegeID
0
2000
4000
6000
8000
10000

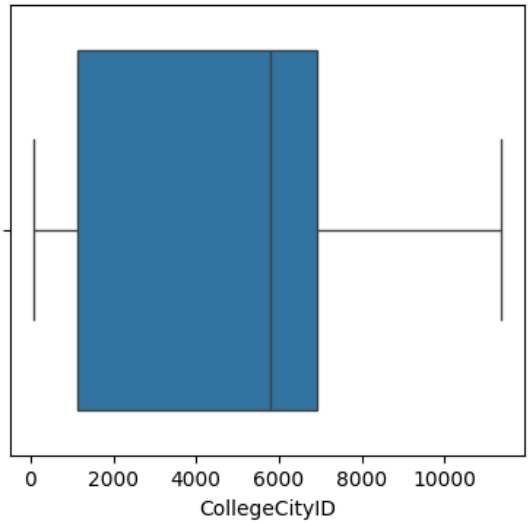

Histogram and Probability Distribution of CollegeTier
Count
CollegeTier
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1.0
1.2
1.4
1.6
1.8
2.0

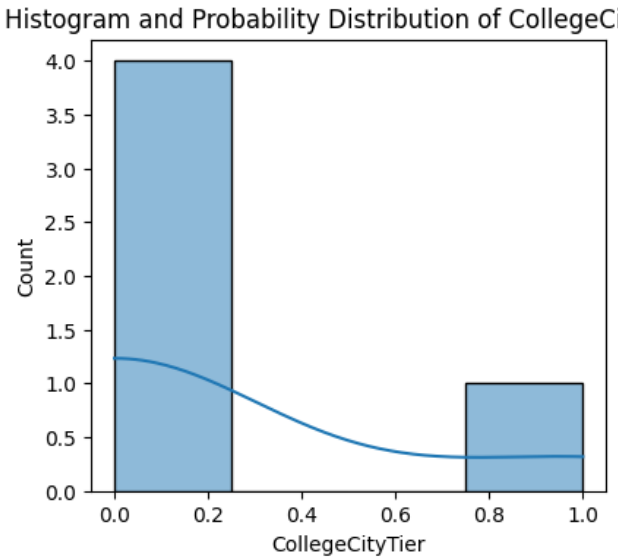

Boxplot of CollegeTier
CollegeTier
1.0
1.2
1.4
1.6
1.8
2.0

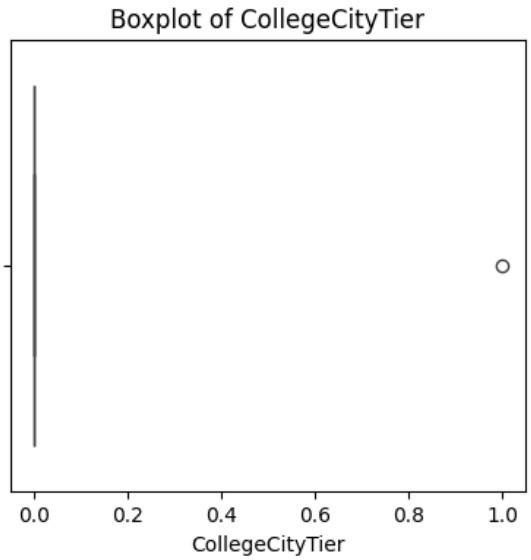

Histogram and Probability Distribution of collegeGPA
Count
collegeGPA
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
70
72
74
76
78

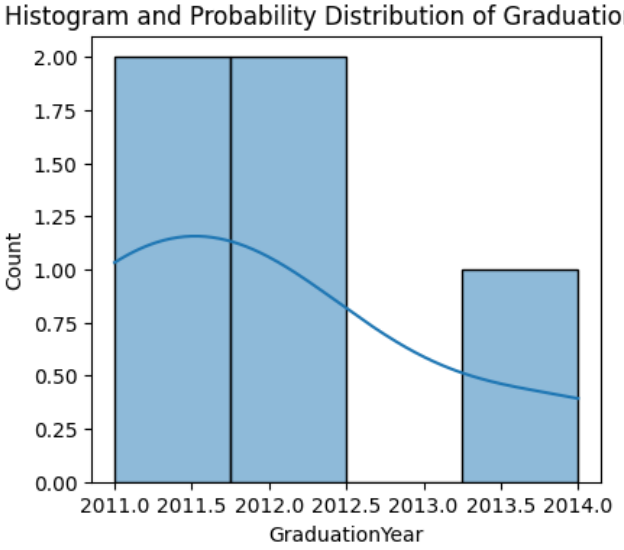

Boxplot of collegeGPA
collegeGPA
70
72
74
76
78

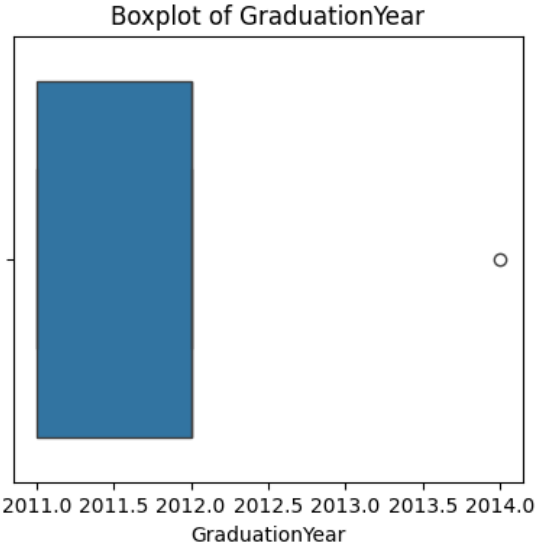

Histogram and Probability Distribution of CollegeCityID
Count
CollegeCityID
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0
2000
4000
6000
8000
10000

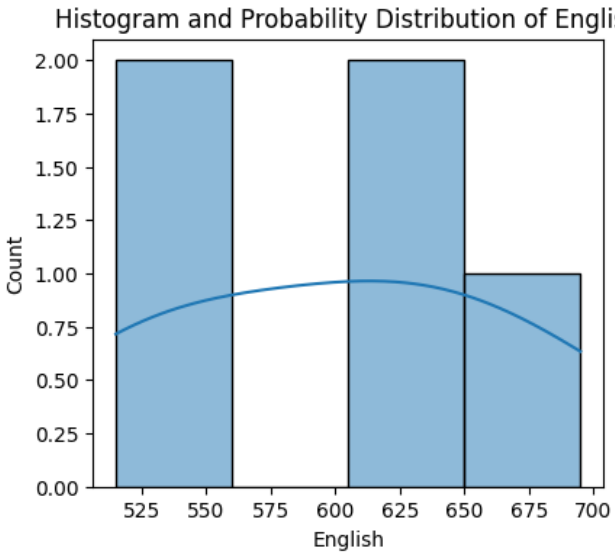

Boxplot of CollegeCityID
CollegeCityID
0
2000
4000
6000
8000
10000

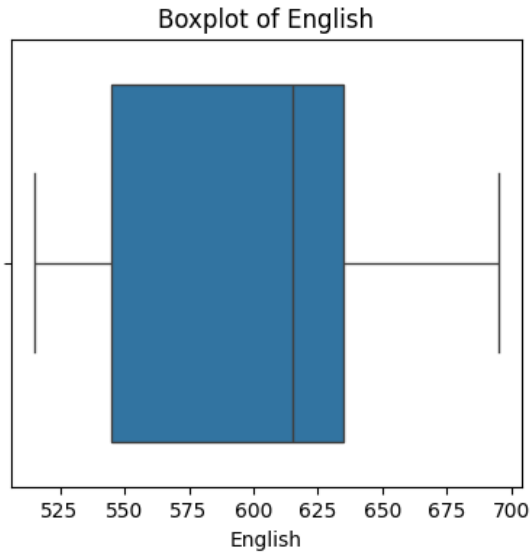

Histogram and Probability Distribution of CollegeCityTier
Count
CollegeCityTier
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0.0
0.2
0.4
0.6
0.8
1.0

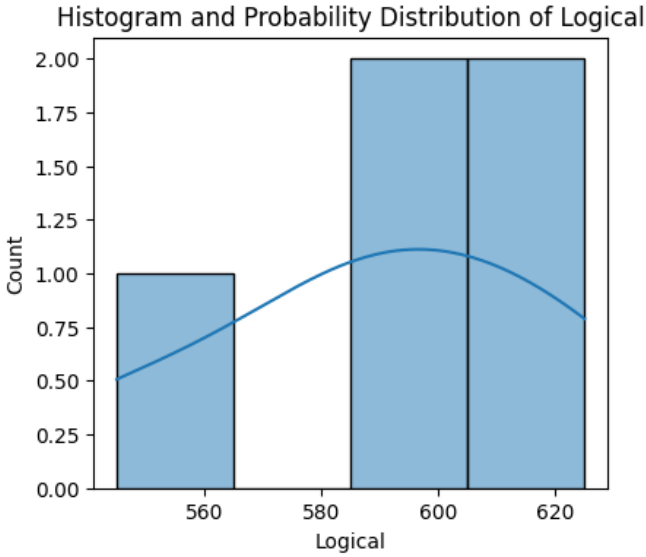

Boxplot of CollegeCityTier
CollegeCityTier
0.0
0.2
0.4
0.6
0.8
1.0

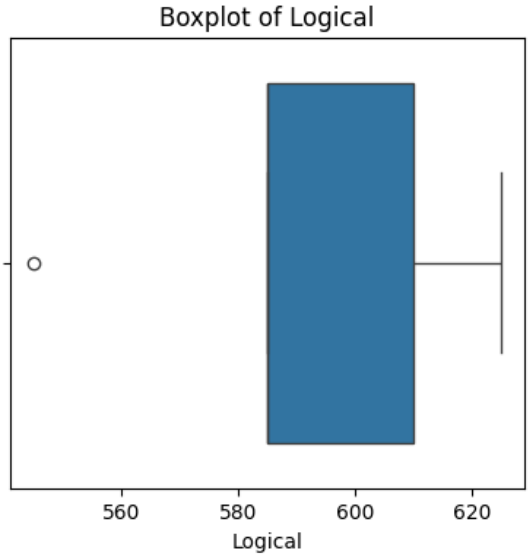

Histogram and Probability Distribution of GraduationYear
Count
GraduationYear
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2011.0
2011.5
2012.0
2012.5
2013.0
2013.5
2014.0

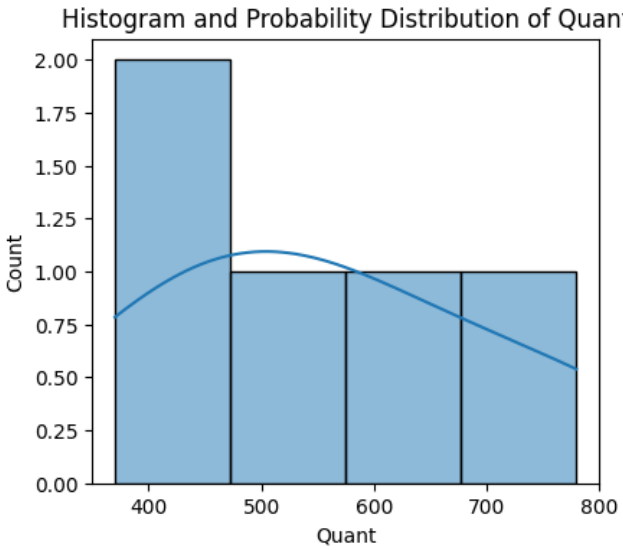

Boxplot of GraduationYear
GraduationYear
2011.0
2011.5
2012.0
2012.5
2013.0
2013.5
2014.0

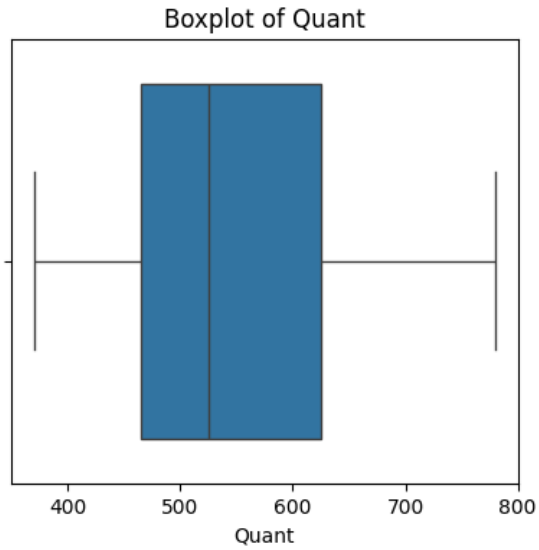

Histogram and Probability Distribution of English
Count
English
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
525
550
575
600
625
650
675
700

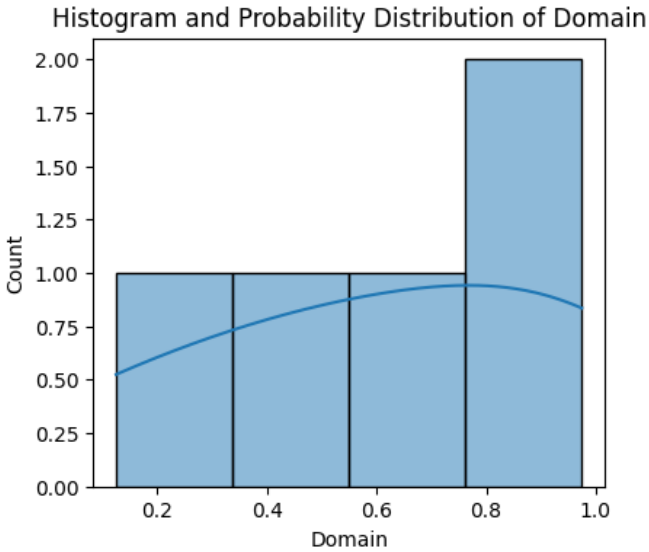

Boxplot of English
English
525
550
575
600
625
650
675
700

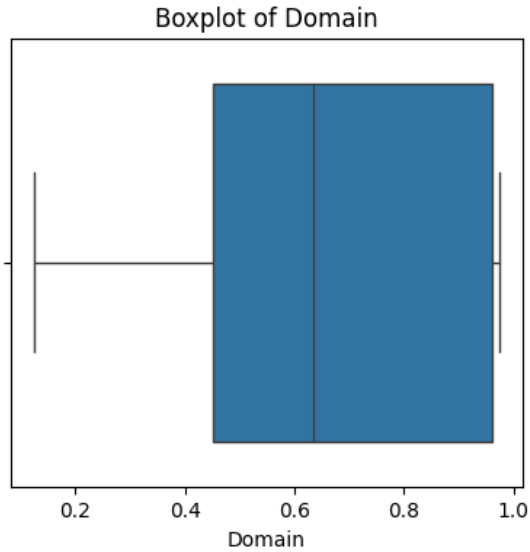

Histogram and Probability Distribution of Logical
Count
Logical
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
560
580
600
620

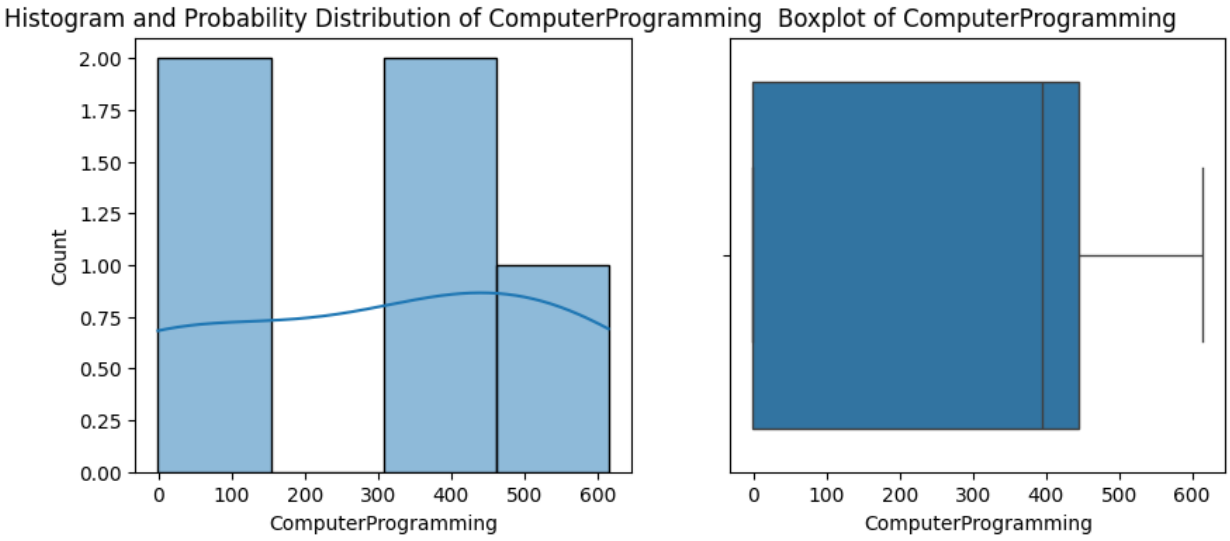

Boxplot of Logical
Logical
560
580
600
620

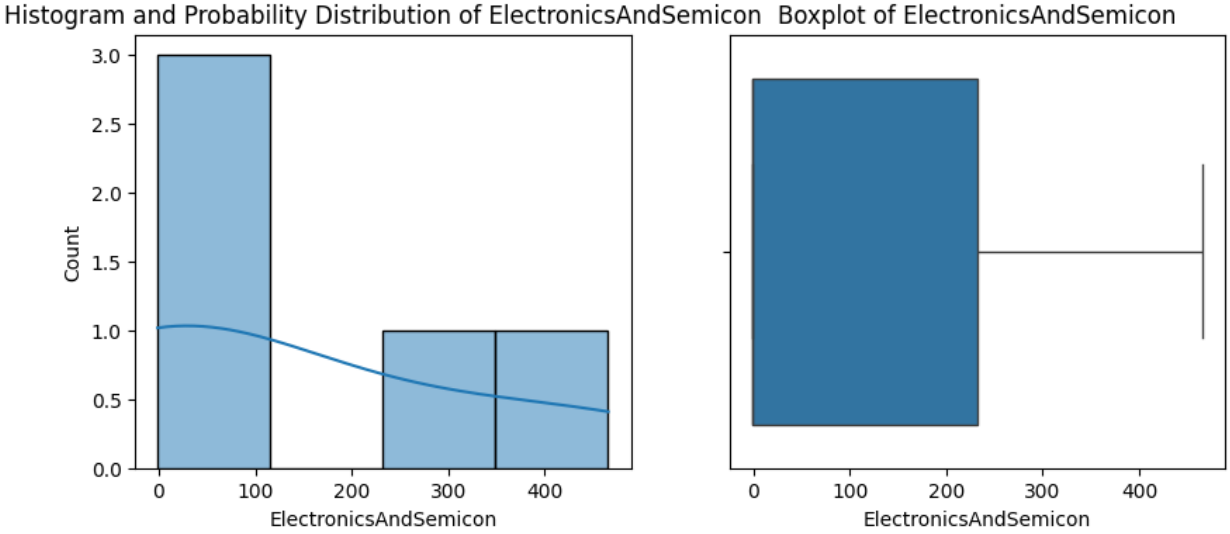

Histogram and Probability Distribution of Quant
Count
Quant
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
400
500
600
700
800

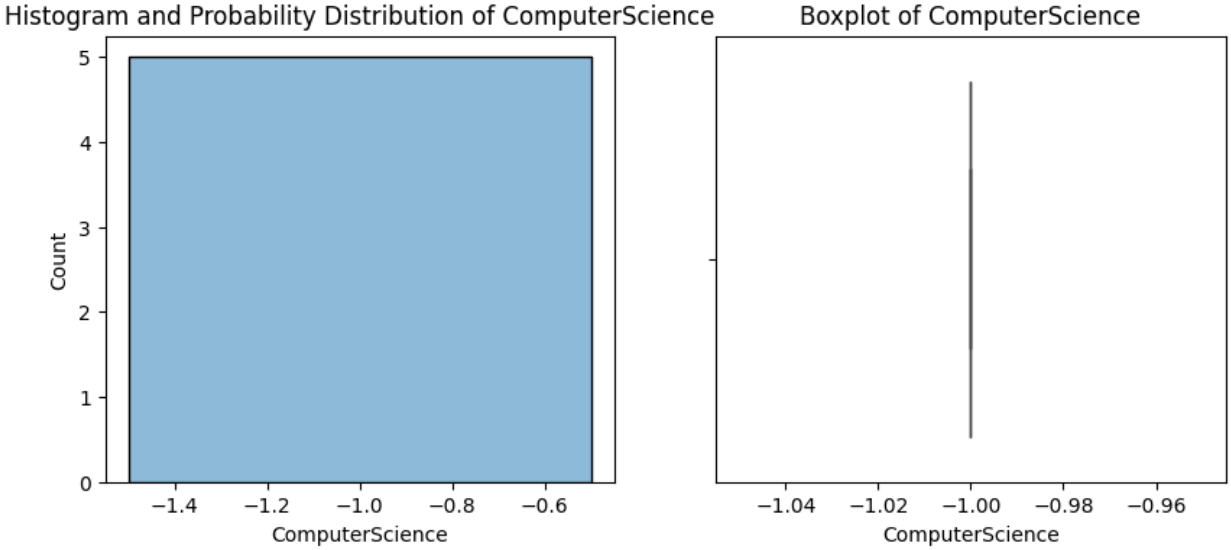

Boxplot of Quant
Quant
400
500
600
700
800

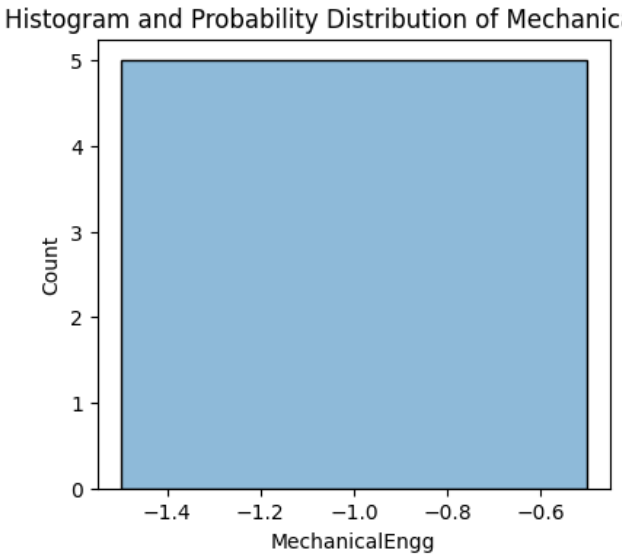

Histogram and Probability Distribution of Domain
Count
Domain
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.2
0.4
0.6
0.8
1.0

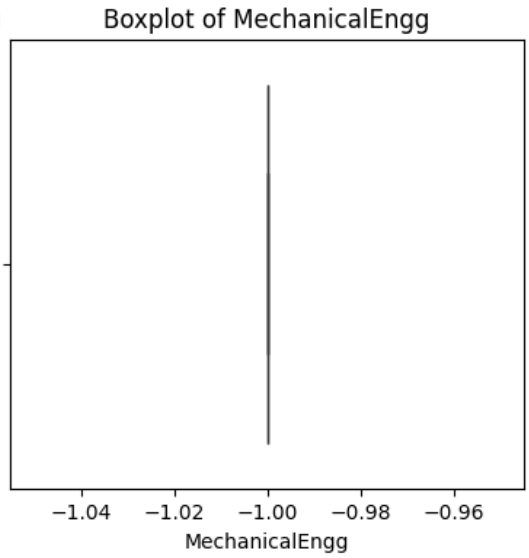

Boxplot of Domain
Domain
0.2
0.4
0.6
0.8
1.0

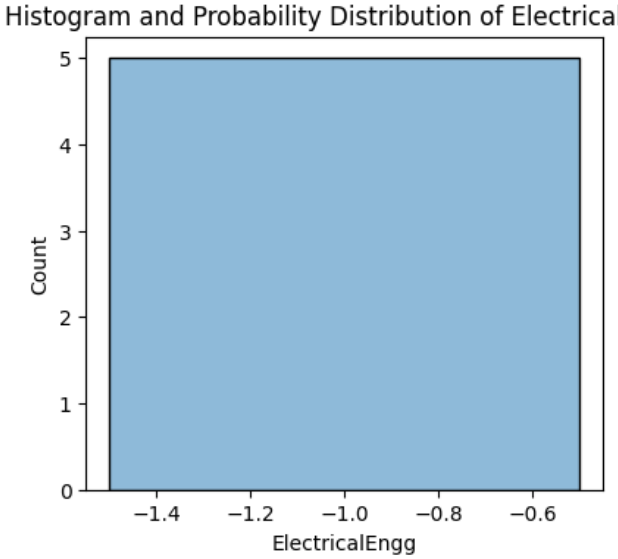

Histogram and Probability Distribution of ComputerProgramming
Count
ComputerProgramming
Boxplot of ComputerProgramming
ComputerProgramming

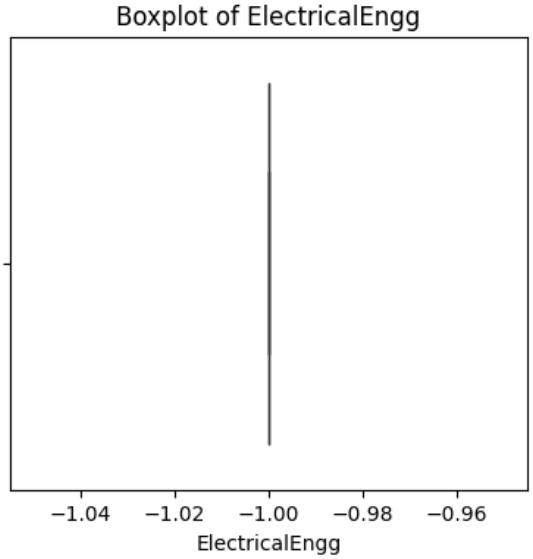

Histogram and Probability Distribution of ElectronicsAndSemicon
Count
ElectronicsAndSemicon
Boxplot of ElectronicsAndSemicon
ElectronicsAndSemicon

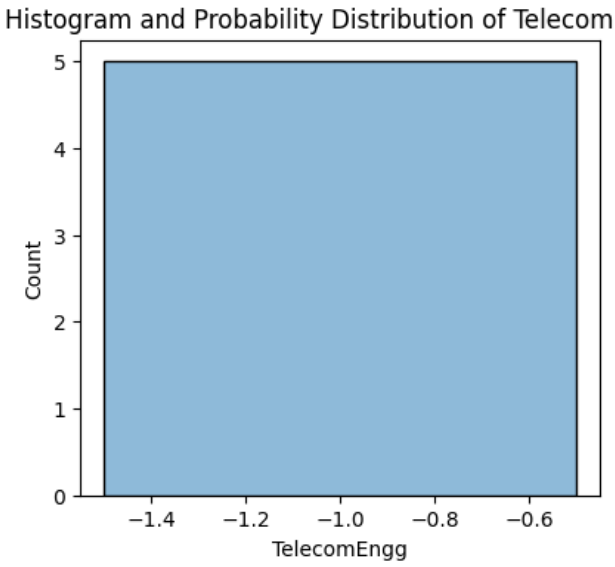

Histogram and Probability Distribution of ComputerScience
Count
ComputerScience
Boxplot of ComputerScience
ComputerScience

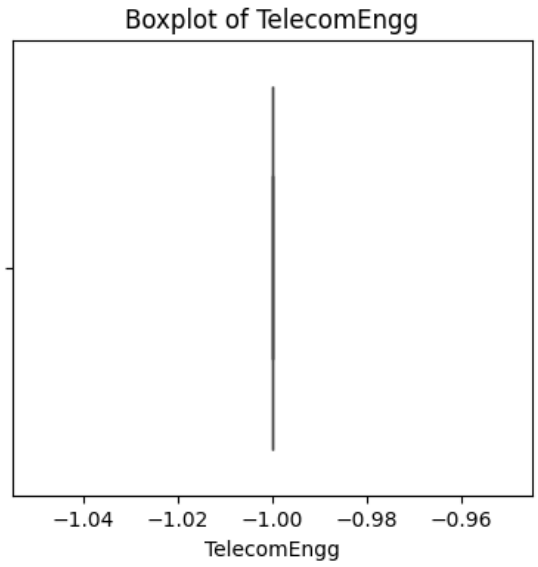

Histogram and Probability Distribution of MechanicalEngg
Count
MechanicalEngg
0
1
2
3
4
5
−1.4
−1.2
−1.0
−0.8
−0.6

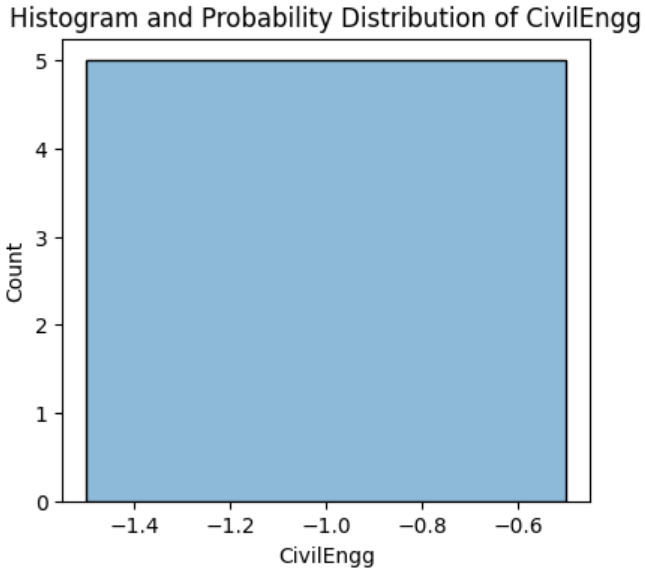

Boxplot of MechanicalEngg
MechanicalEngg
−1.04
−1.02
−1.00
−0.98
−0.96

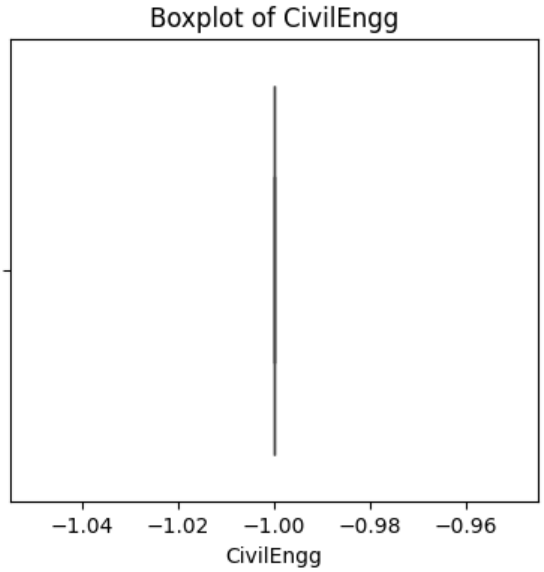

Histogram and Probability Distribution of ElectricalEngg
Count
ElectricalEngg
0
1
2
3
4
5
−1.4
−1.2
−1.0
−0.8
−0.6

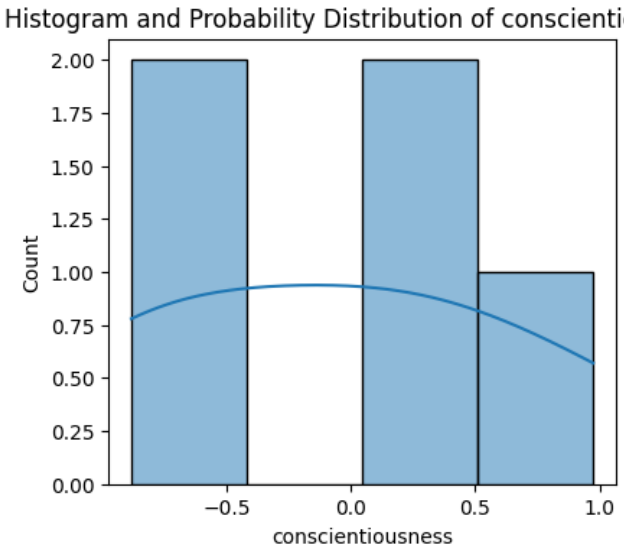

Boxplot of ElectricalEngg
ElectricalEngg
−1.04
−1.02
−1.00
−0.98
−0.96

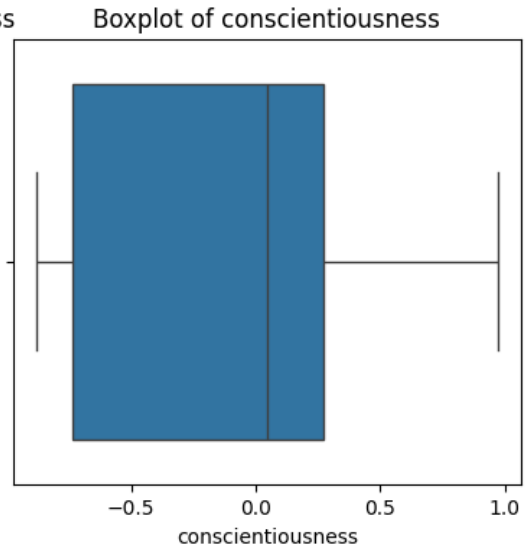

Histogram and Probability Distribution of TelecomEngg
Count
TelecomEngg
0
1
2
3
4
5
−1.4
−1.2
−1.0
−0.8
−0.6

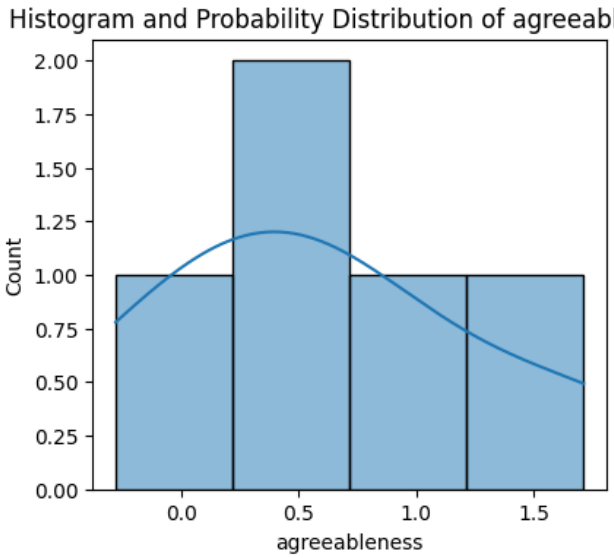

Boxplot of TelecomEngg
TelecomEngg
−1.04
−1.02
−1.00
−0.98
−0.96

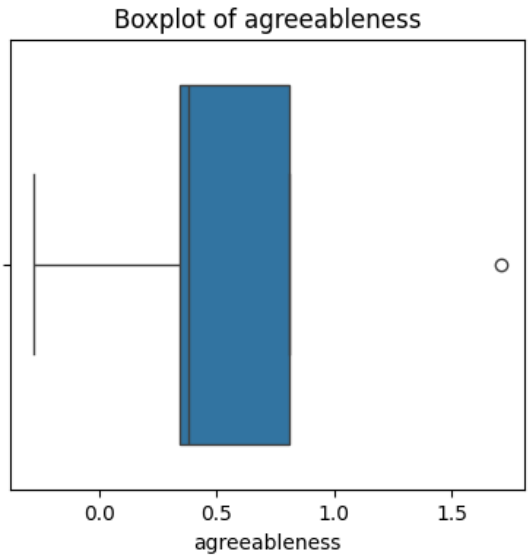

Histogram and Probability Distribution of CivilEngg
Count
CivilEngg
0
1
2
3
4
5
−1.4
−1.2
−1.0
−0.8
−0.6

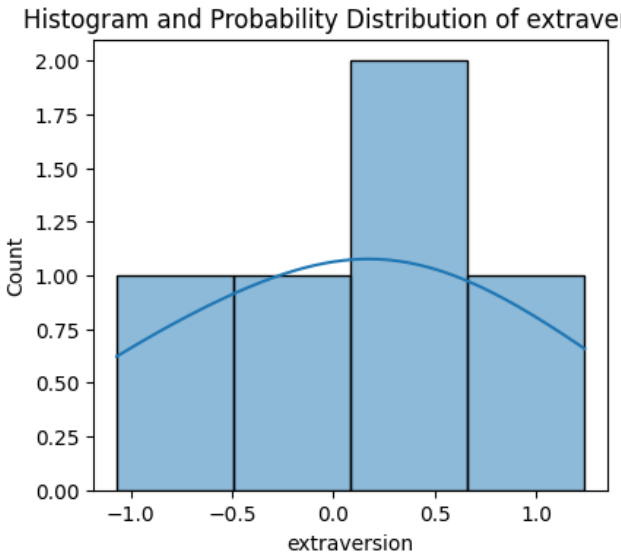

Boxplot of CivilEngg
CivilEngg
−1.04
−1.02
−1.00
−0.98
−0.96

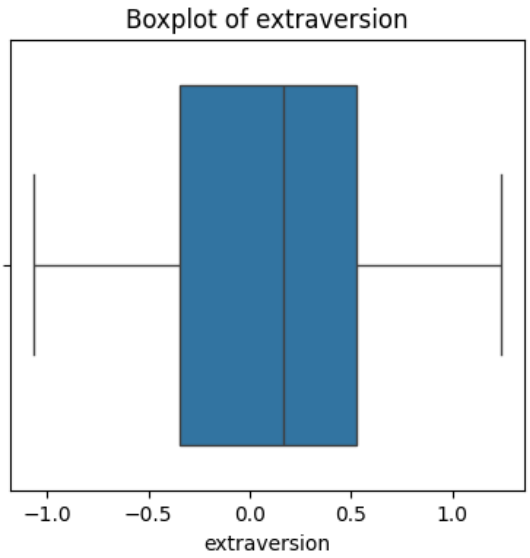

Histogram and Probability Distribution of conscientiousness
Count
conscientiousness
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.5
0.0
0.5
1.0

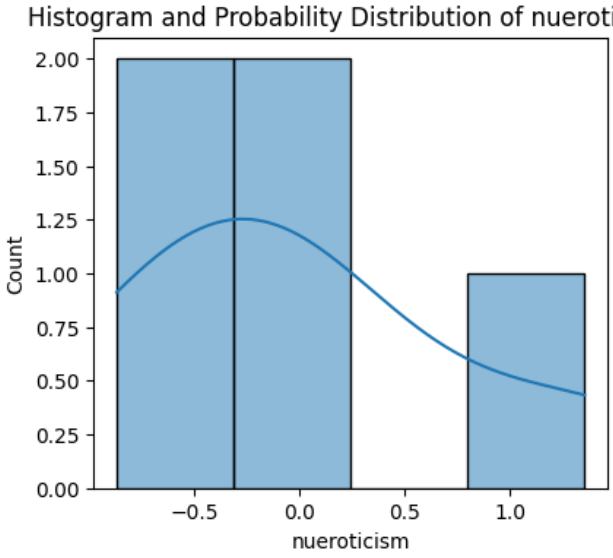

Boxplot of conscientiousness
conscientiousness
−0.5
0.0
0.5
1.0

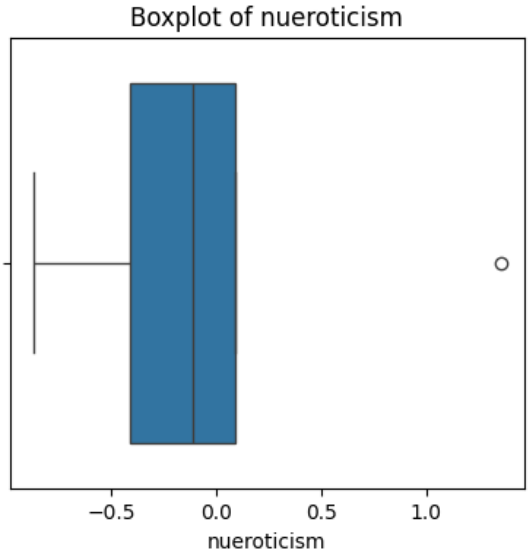

Histogram and Probability Distribution of agreeableness
Count
agreeableness
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
1.5

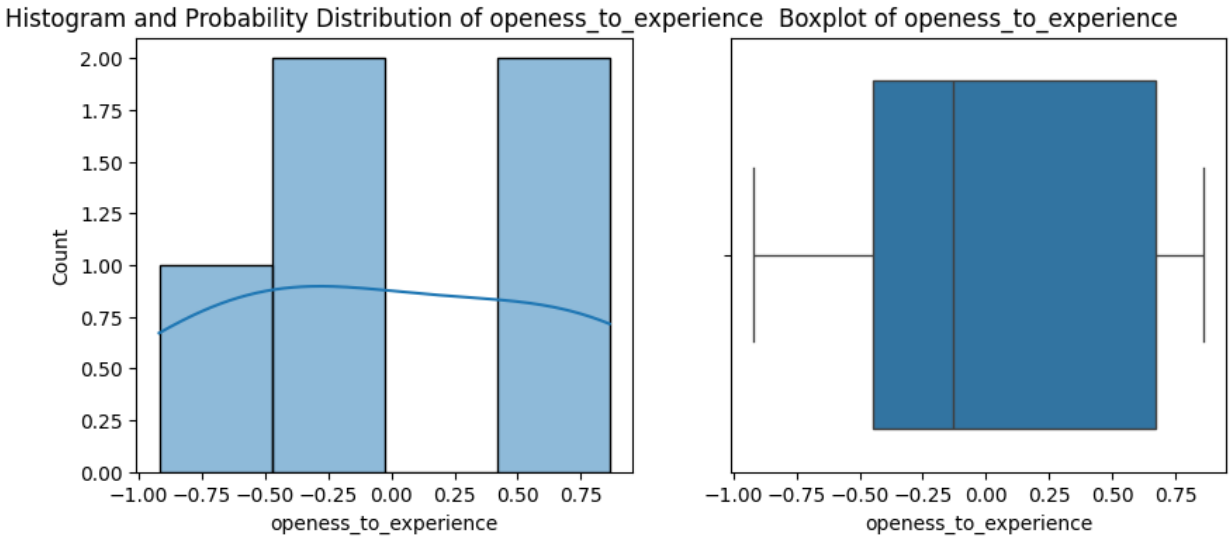

Boxplot of agreeableness
agreeableness
0.0
0.5
1.0
1.5

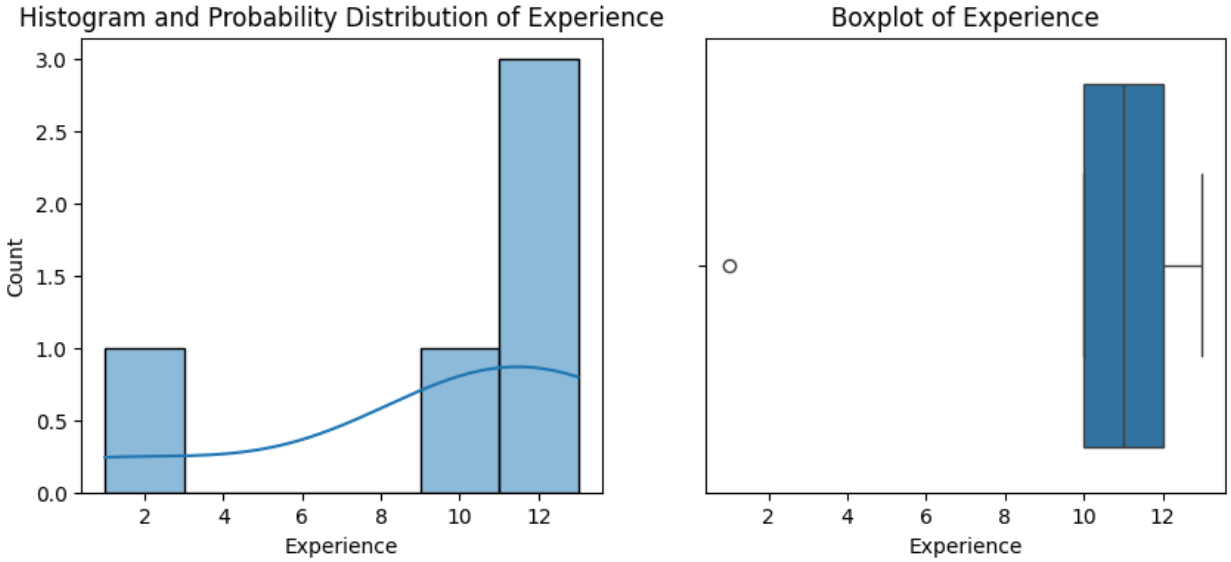

Histogram and Probability Distribution of extraversion
Count
extraversion
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.0
−0.5
0.0
0.5
1.0


Boxplot of extraversion
extraversion
−1.0
−0.5
0.0
0.5
1.0

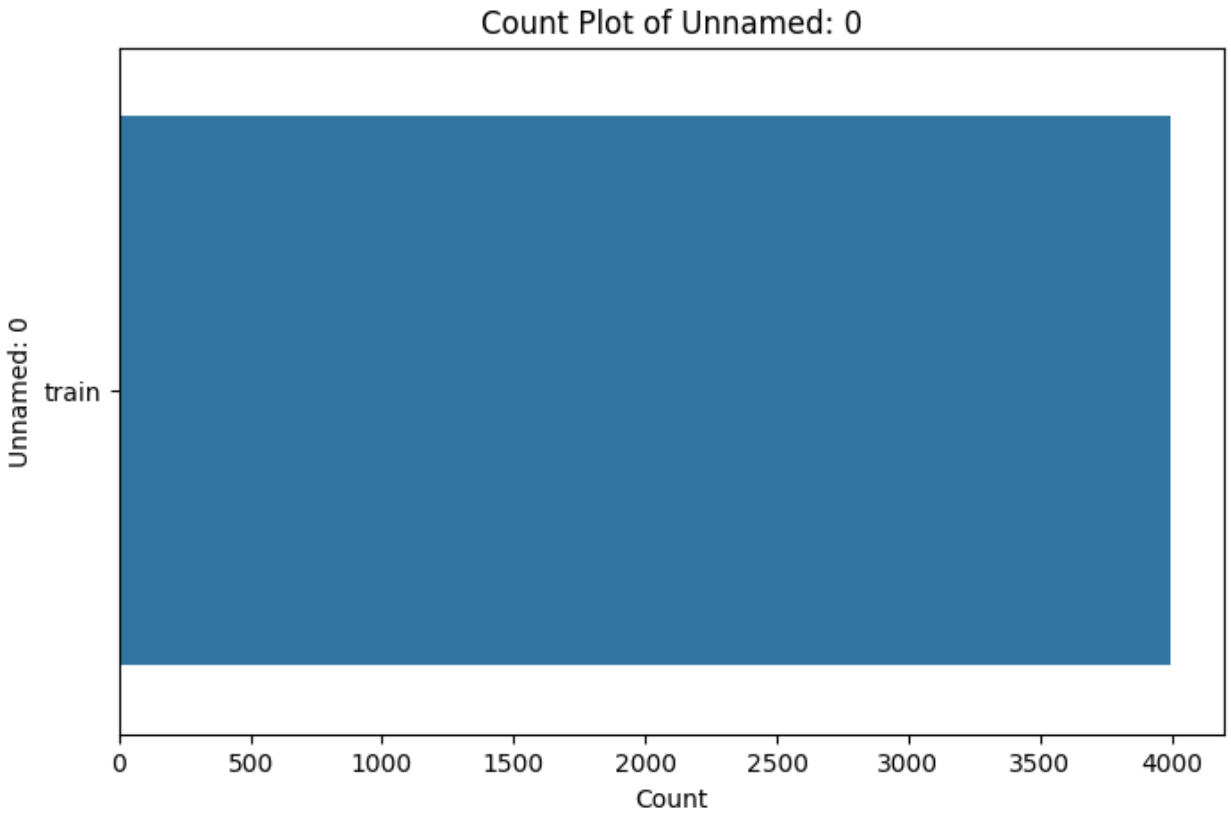

Histogram and Probability Distribution of nueroticism
Count
nueroticism
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.5
0.0
0.5
1.0

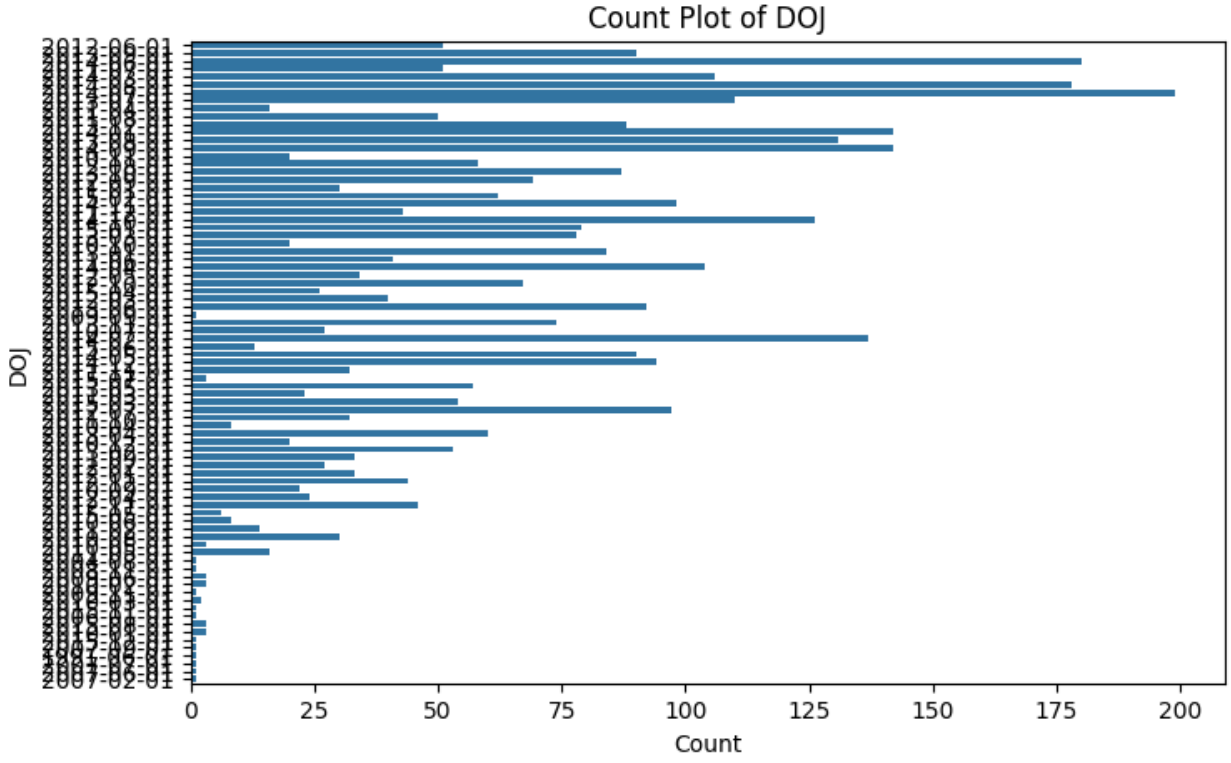

Boxplot of nueroticism
nueroticism
−0.5
0.0
0.5
1.0

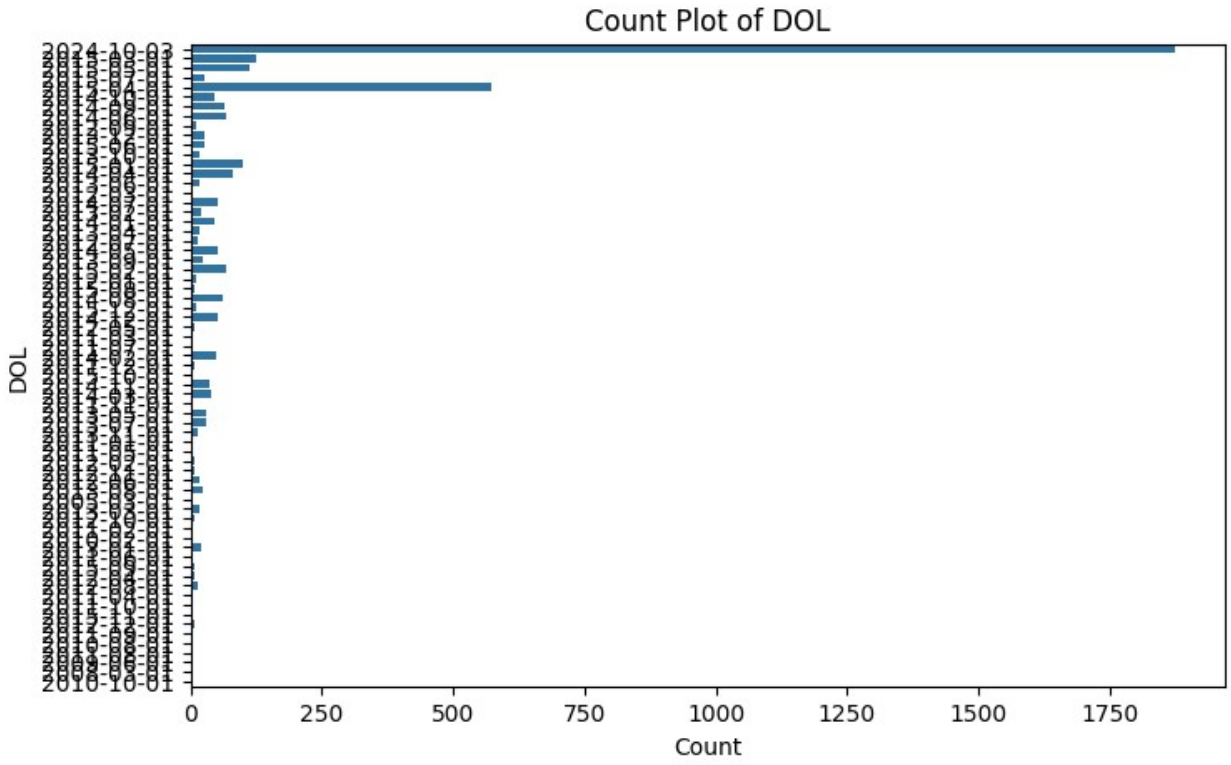


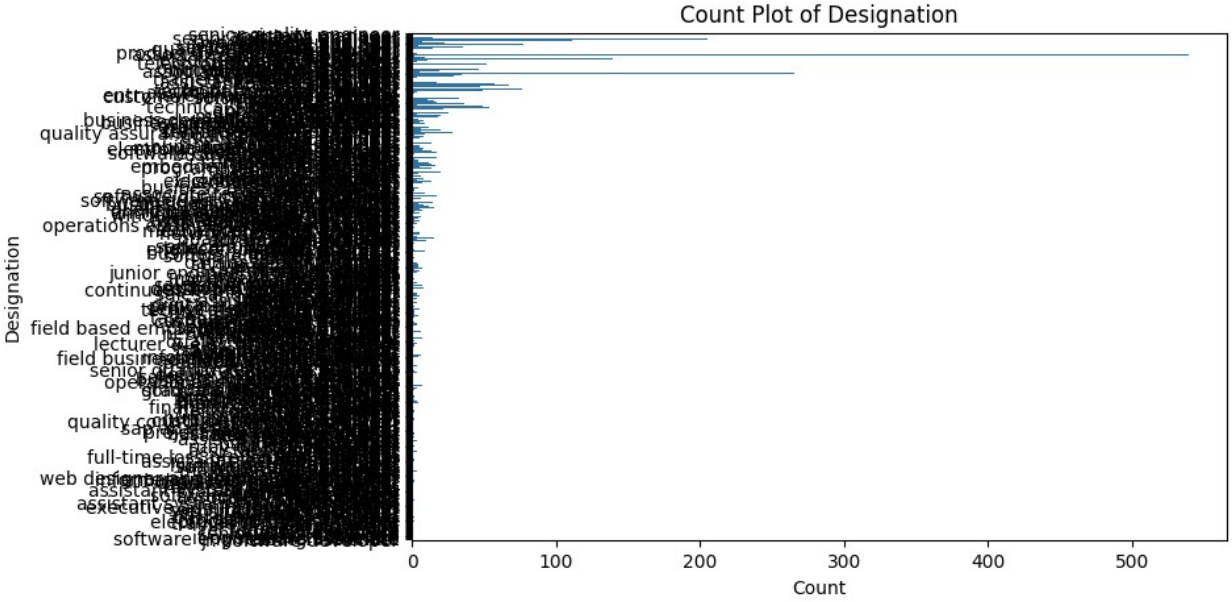


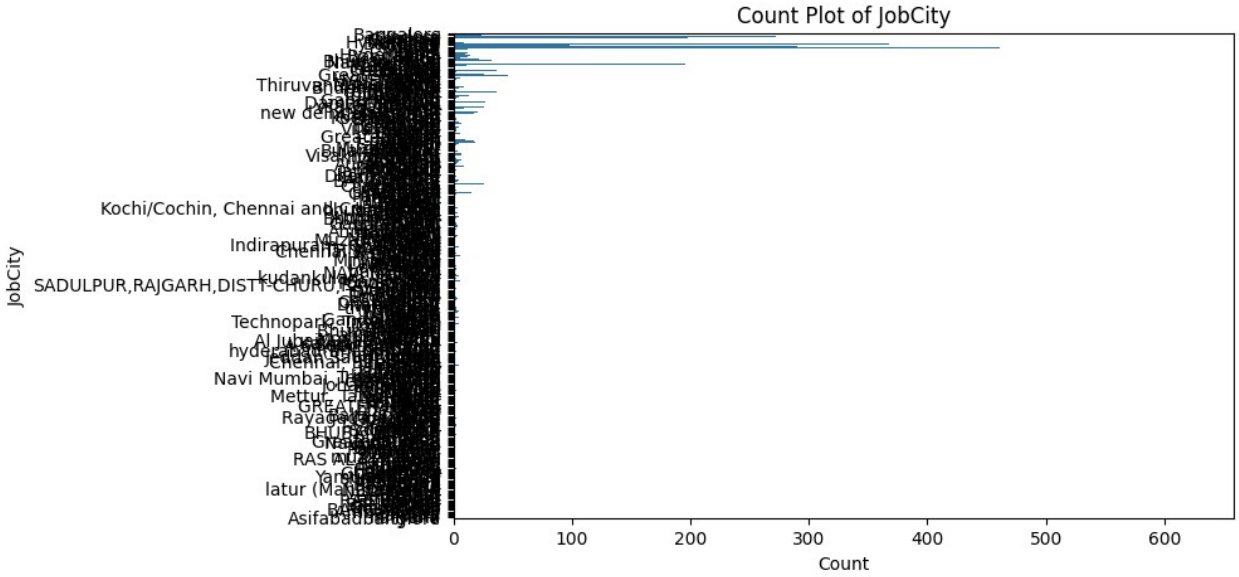

```
# Plotting count plots for categorical variables
cat_col = df.select_dtypes(include=['object']).columns
for column in cat_col:
    plt.figure(figsize=(8, 5))
    sns.countplot(y=df[column])
    plt.title(f'Count Plot of {column}')
    plt.xlabel('Count')
    plt.ylabel(column)
    plt.show()
```

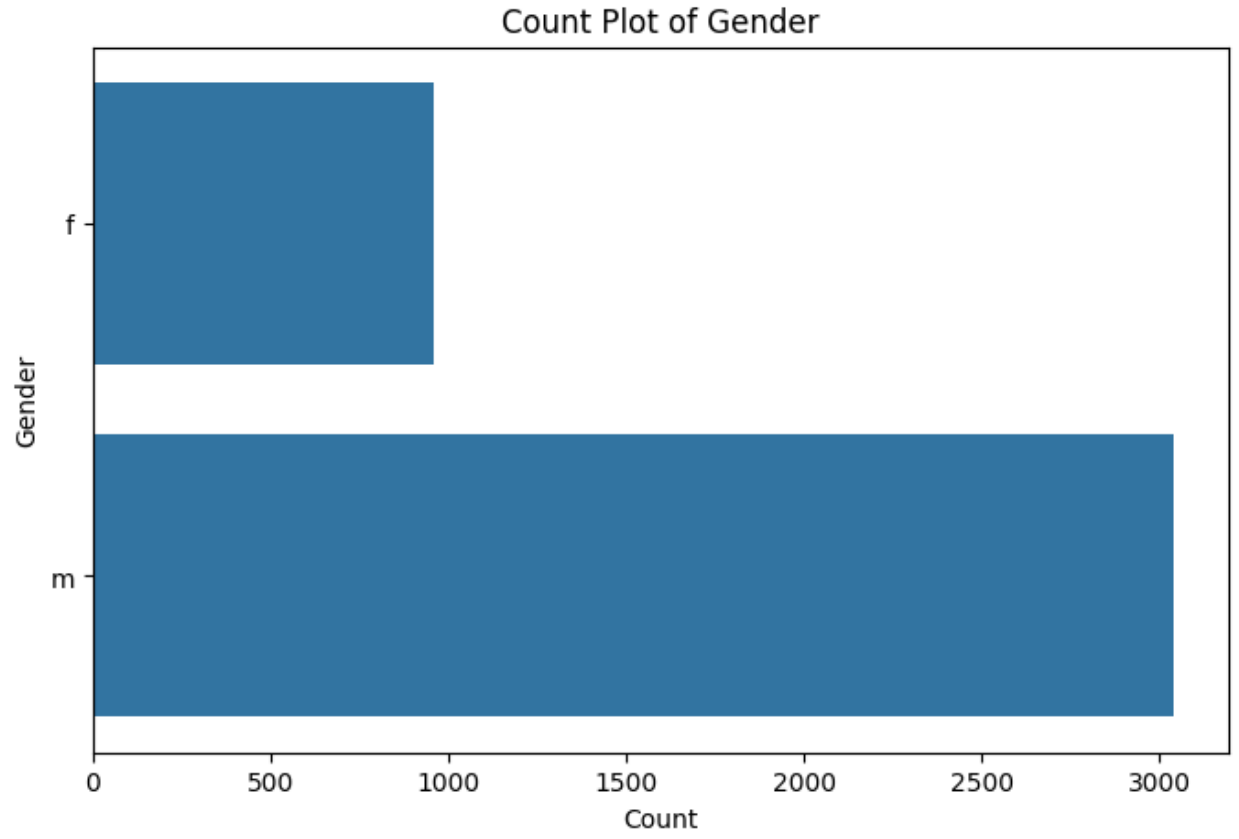

Count Plot of Unnamed: 0
Unnamed: 0
train
0
500
1000
1500
2000
2500
3000
3500
4000
Count

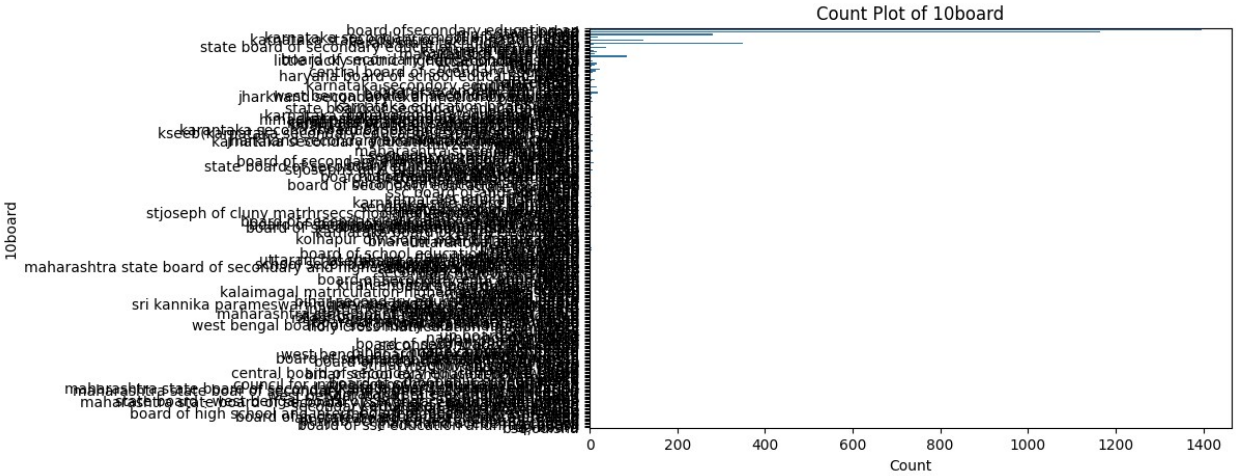

Count Plot of DOJ
DOJ
0
25
50
75
100
125
150
175
200
Count

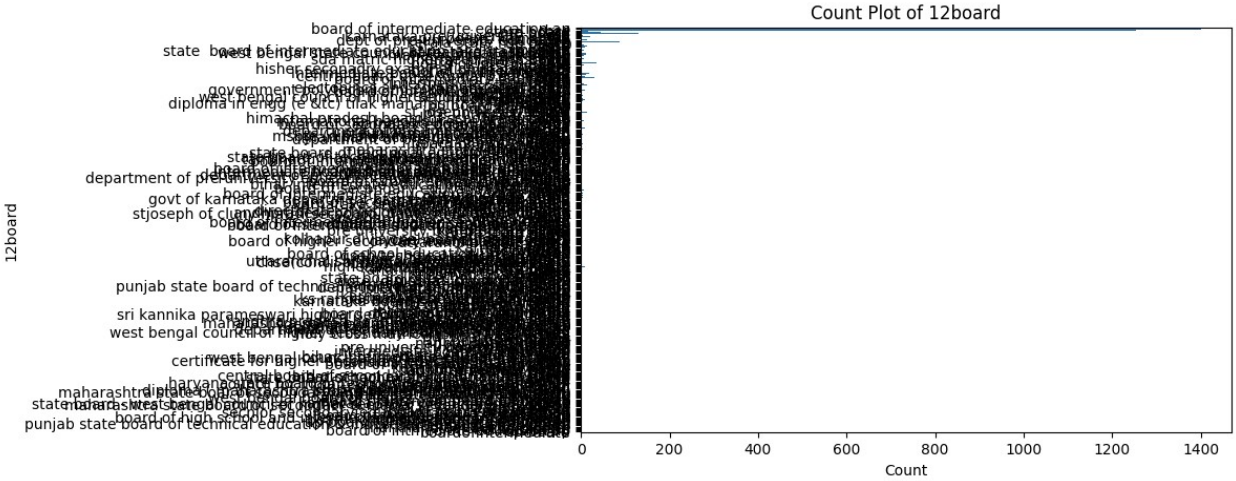

Count Plot of DOL
DOL
2024-10-03
2010-10-01
0
250
500
750
1000
1250
1500
1750
Count

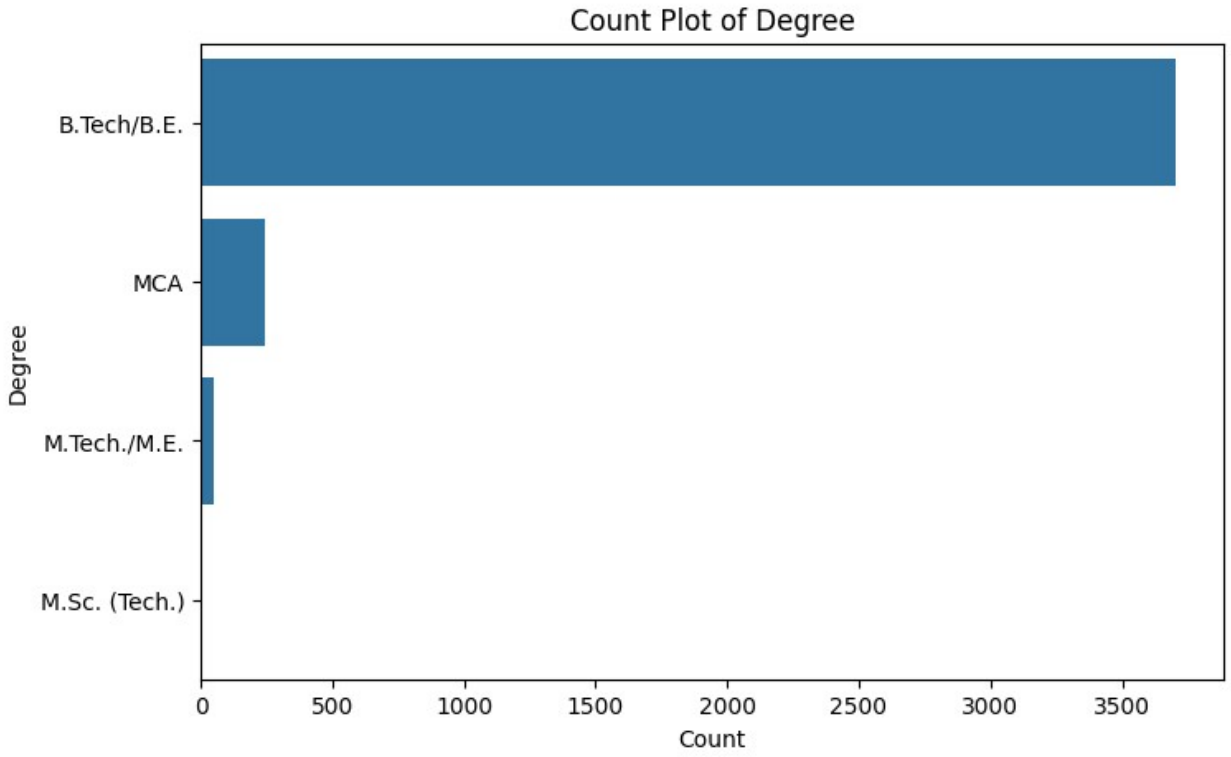

Count Plot of Designation
Designation
0
100
200
300
400
500
Count

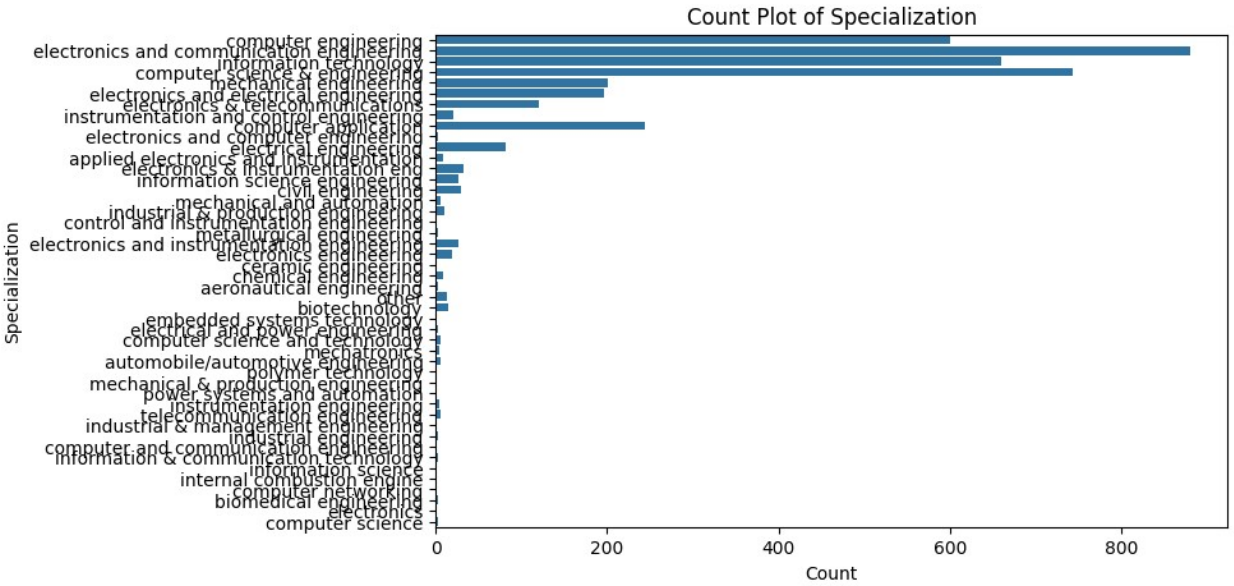

Count Plot of JobCity
JobCity
Count
0
100
200
300
400
500
600

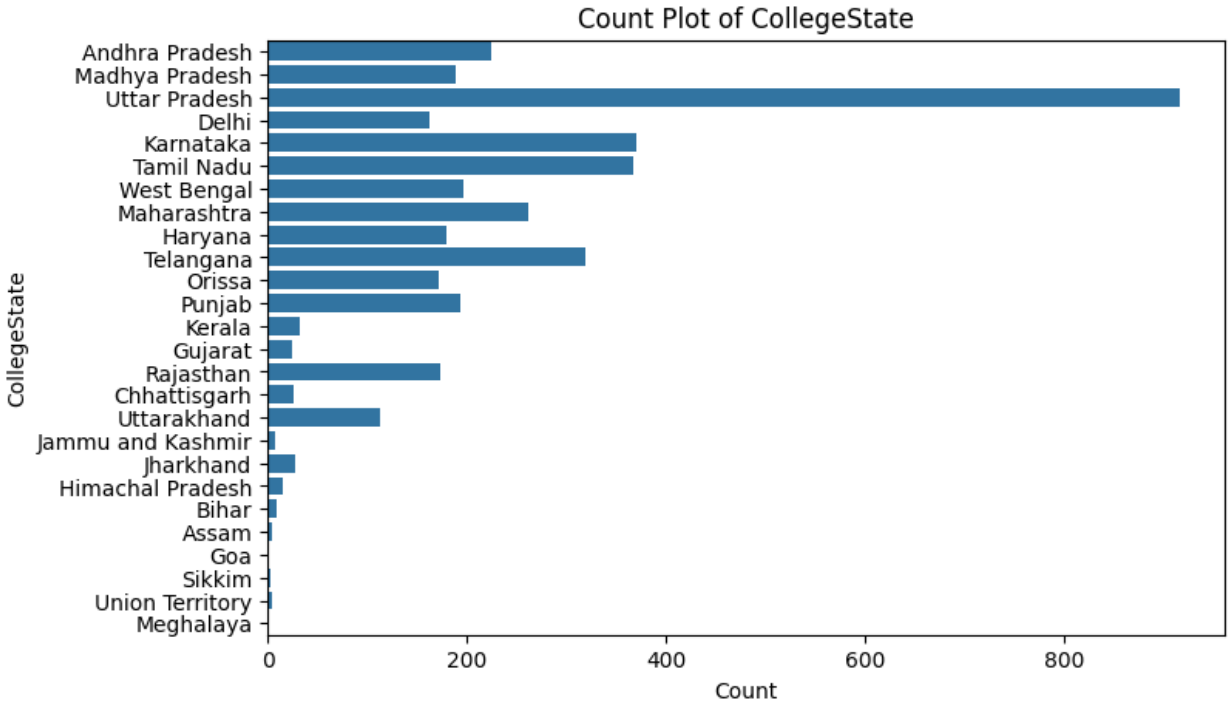

Count Plot of Gender
Gender
f
m
Count
0
500
1000
1500
2000
2500
3000


Count Plot of 10board
10board
Count
0
200
400
600
800
1000
1200
1400


Count Plot of 12board
12board
Count
0
200
400
600
800
1000
1200
1400


Count Plot of Degree
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)
Degree
0
500
1000
1500
2000
2500
3000
3500
Count


Count Plot of Specialization
computer engineering
electronics and communication engineering
information technology
computer science & engineering
mechanical engineering
electronics and electrical engineering
electronics & telecommunications
instrumentation and control engineering
computer application
electronics and computer engineering
electrical engineering
applied electronics and instrumentation
electronics & instrumentation eng
information science engineering
civil engineering
mechanical and automation
industrial & production engineering
control and instrumentation engineering
metallurgical engineering
electronics and instrumentation engineering
electronics engineering
ceramic engineering
chemical engineering
aeronautical engineering
other
biotechnology
embedded systems technology
electrical and power engineering
computer science and technology
mechatronics
automobile/automotive engineering
polymer technology
mechanical & production engineering
power systems and automation
instrumentation engineering
telecommunication engineering
industrial & management engineering
industrial engineering
computer and communication engineering
information & communication technology
information science
internal combustion engine
computer networking
biomedical engineering
electronics
computer science
Specialization
0
200
400
600
800
Count




```
# Correlation between Salary and other numerical columns
correlation = df1[num_columns].corr()
print("Correlation with Salary:\n",
correlation['Salary'].sort_values(ascending=False))
```

```
Correlation with Salary:
 Salary                    1.000000
CollegeCityTier            0.947213
Domain                     0.777012
Experience                 0.648456
ComputerProgramming        0.647491
10percentage               0.576554
Quant                      0.473860
English                    0.414588
12percentage               0.268541
collegeGPA                 0.174992
conscientiousness          0.159690
CollegeCityID              0.064302
CollegeID                  0.064302
extraversion               0.032062
agreeableness             -0.097858
Logical                   -0.123138
nueroticism               -0.197230
ElectronicsAndSemicon     -0.262506
ID                        -0.344118
12graduation              -0.473319
GraduationYear            -0.509153
openess_to_experience     -0.567771
```

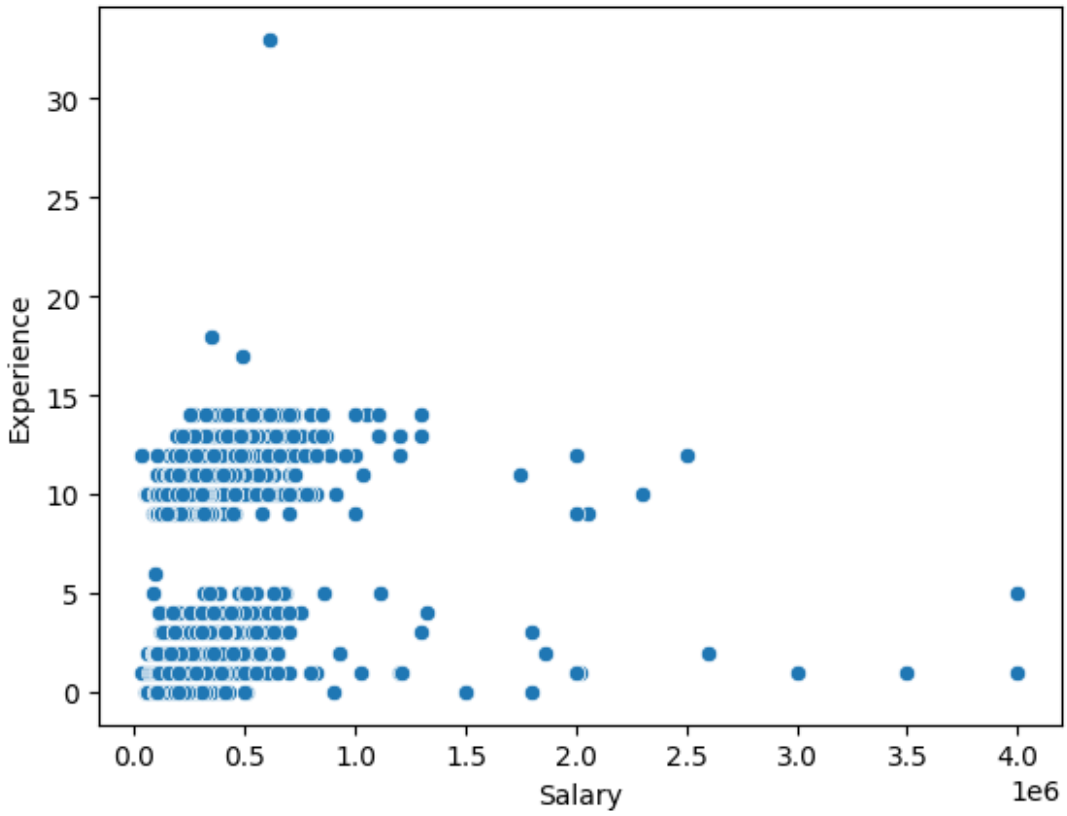
```
DOB                       -0.692112
CollegeTier               -0.947213
ComputerScience                 NaN
MechanicalEngg                  NaN
ElectricalEngg                  NaN
TelecomEngg                     NaN
CivilEngg                       NaN
Name: Salary, dtype: float64
```

Bivariate Analysis

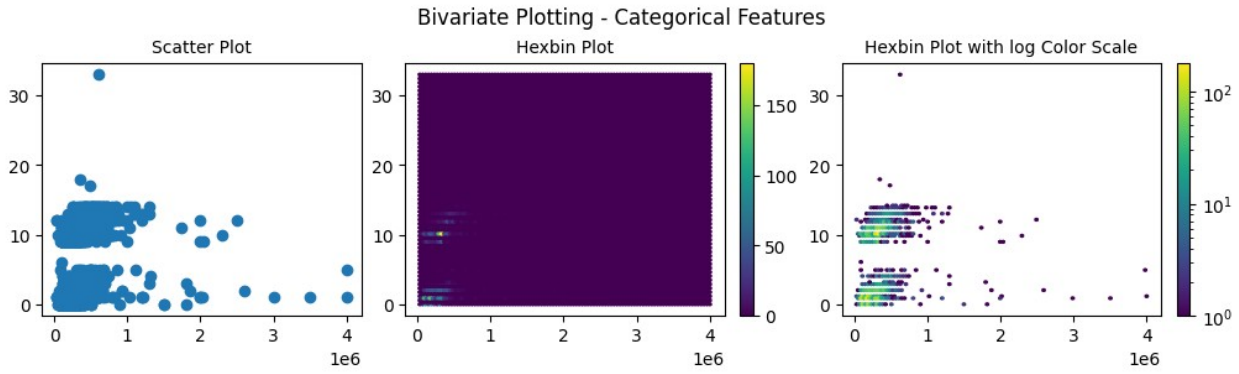
- Discover the relationships between numerical columns using Scatter plots, hexbin plots, pair plots, etc..
- Identify the patterns between categorical and numerical columns using swarmplot, boxplot, barplot, etc..
- Identify relationships between categorical and categorical columns using stacked bar plots.
- Mention observations after each plt.

```
sns.scatterplot(data=df, x='Salary', y='Experience')
print("Scatter plot for Salary vs Experience to show the direct relationship btw. salary will increases with the experience. trend upwards, experience incerases , salary trend to rise")
```

```
Scatter plot for Salary vs Experience to show the direct relationship btw. salary will increases with the experience. trend upwards, experience incerases , salary trend to rise
```




```
fig, axs = plt.subplots(1, 3, figsize=(10, 3), layout="constrained")
fig.suptitle("Bivariate Plotting - Categorical Features")

axs[0].scatter(df["Salary"], df["Experience"])
axs[0].set_title("Scatter Plot", fontsize="medium")

hb1 = axs[1].hexbin(df["Salary"], df["Experience"])
axs[1].set_title("Hexbin Plot", fontsize="medium")
fig.colorbar(hb1, ax=axs[1])

hb2 = axs[2].hexbin(df["Salary"], df["Experience"], bins="log")
axs[2].set_title("Hexbin Plot with log Color Scale",
fontsize="medium")
fig.colorbar(hb2, ax=axs[2])
plt.show()
print("Scatter plot for Salary vs Experience to show the direct
relationship btw. salary will increases with the experience. trend
upwards, experience incerases , salary trend to rise")
print("Hebix plot for a more granular view , with the density of
points better visualized & Log Color Scale for highlights high density
regions more clear")
```



Scatter plot for Salary vs Experience to show the direct relationship btw. salary will increases with the experience. trend upwards, experience incerases , salary trend to rise
Hebix plot for a more granular view , with the density of points better visualized & Log Color Scale for highlights high density regions more clear

```
print(df['Salary'].value_counts().head(15))
print(df['Specialization'].value_counts().head(15))
print(df['Domain'].value_counts().head(15))
```

```
Salary
300000.0    293
180000.0    239
200000.0    205
325000.0    188
120000.0    165
240000.0    158
400000.0    130
350000.0    125
100000.0    111
150000.0     87
360000.0     75
320000.0     74
450000.0     67
145000.0     64
500000.0     63
Name: count, dtype: int64
Specialization
electronics and communication engineering      880
computer science & engineering                 744
information technology                         660
computer engineering                           600
computer application                           244
mechanical engineering                         201
electronics and electrical engineering         196
electronics & telecommunications               121
electrical engineering                          82
```

```
electronics & instrumentation eng                 32
civil engineering                                 29
electronics and instrumentation engineering       27
information science engineering                   27
instrumentation and control engineering           20
electronics engineering                           19
Name: count, dtype: int64
Domain
-1.000000    246
 0.622643    113
 0.538387    110
 0.486747    106
 0.744758    103
 0.376060    103
 0.356536    102
 0.694479     96
 0.824666     82
 0.229482     81
 0.600057     77
 0.842248     75
 0.735796     70
 0.864685     69
 0.338786     66
Name: count, dtype: int64
```

Scatter Plot

- Scatter plot for Salary vs Experience to show the direct relationship btw. salary will increases with the experience. trend upwards, experience incerases , salary trend to rise.

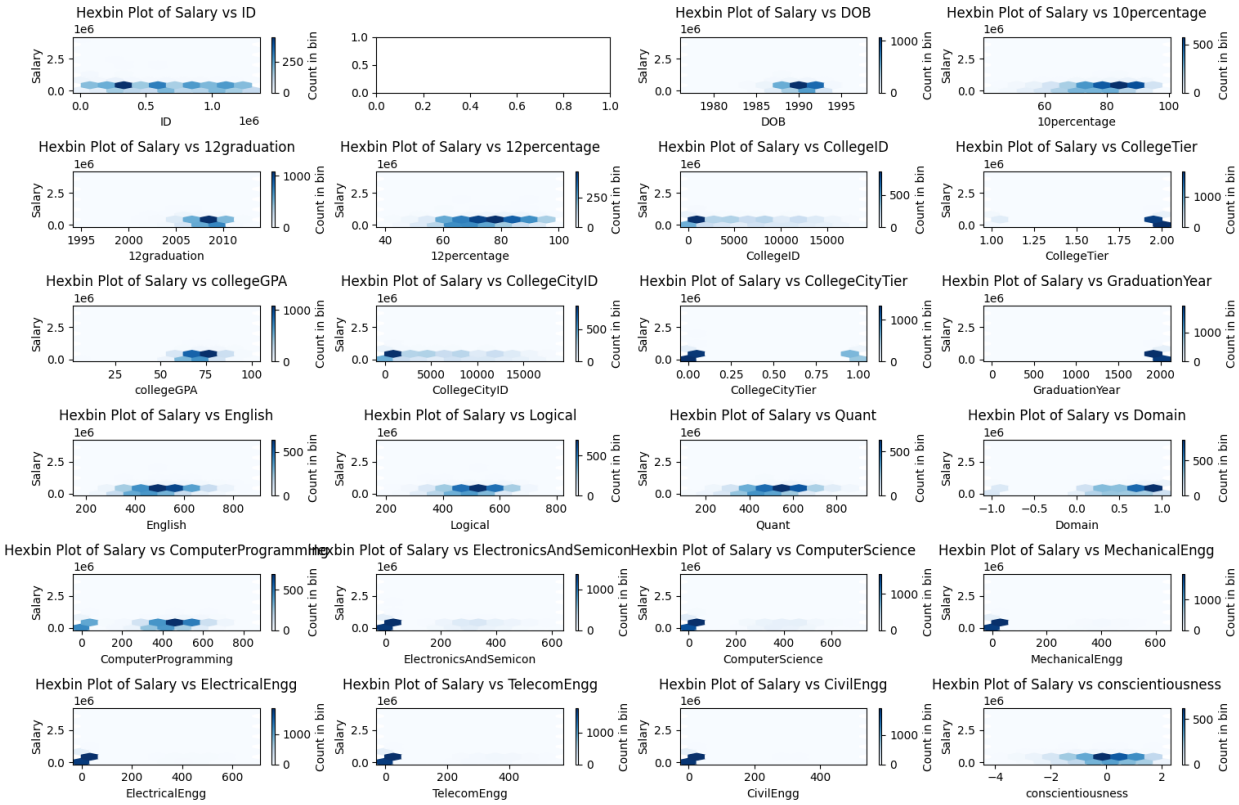
```
# Create a 1x2 subplot layout (1 row, 2 columns)
fig, axs = plt.subplots(1, 2, figsize=(10, 6))

# Scatter Plot for Salary vs Specialization
axs[0].scatter(df['Salary'], df['Specialization'], alpha=0.6)
axs[0].set_title('Scatter Plot of Salary vs Specialization')
axs[0].set_xlabel('Salary')
axs[0].set_ylabel('Specialization')
axs[0].grid()

# Scatter Plot for Salary vs Domain
axs[1].scatter(df['Salary'], df['Domain'], alpha=0.6)
axs[1].set_title('Scatter Plot of Salary vs Domain')
axs[1].set_xlabel('Salary')
axs[1].set_ylabel('Domain')
axs[1].grid()

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

Hexbin plots for a more granular view

1. Hexbin plots enhance scatter plots by aggregating data points into hexagonal bins.
2. Hebix plot for a more granular view , with the density of points better visualized.

```
# Create the subplot layout
fig, axs = plt.subplots(6, 4, figsize=(15, 10))
# Flatten the axes array for easier iteration
axs = axs.flatten()
# Loop through the numerical columns except 'Salary'
for idx, column in enumerate(num_columns):
    if column != 'Salary':
        if idx < len(axs):  # Ensure idx is within bounds of axs
            # Plot each hexbin in the corresponding subplot
            hb = axs[idx].hexbin(df[column], df['Salary'],
gridsize=10, cmap='Blues')
            axs[idx].set_title(f'Hexbin Plot of Salary vs {column}')
            axs[idx].set_xlabel(column)
            axs[idx].set_ylabel('Salary')

            # Add a colorbar associated with the hexbin plot
            fig.colorbar(hb, ax=axs[idx], label='Count in bin')
# Remove any empty subplots if the number of columns is less than rows
* columns
for i in range(len(num_columns), len(axs)):
    fig.delaxes(axs[i])
# Adjust layout to prevent overlap
```

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

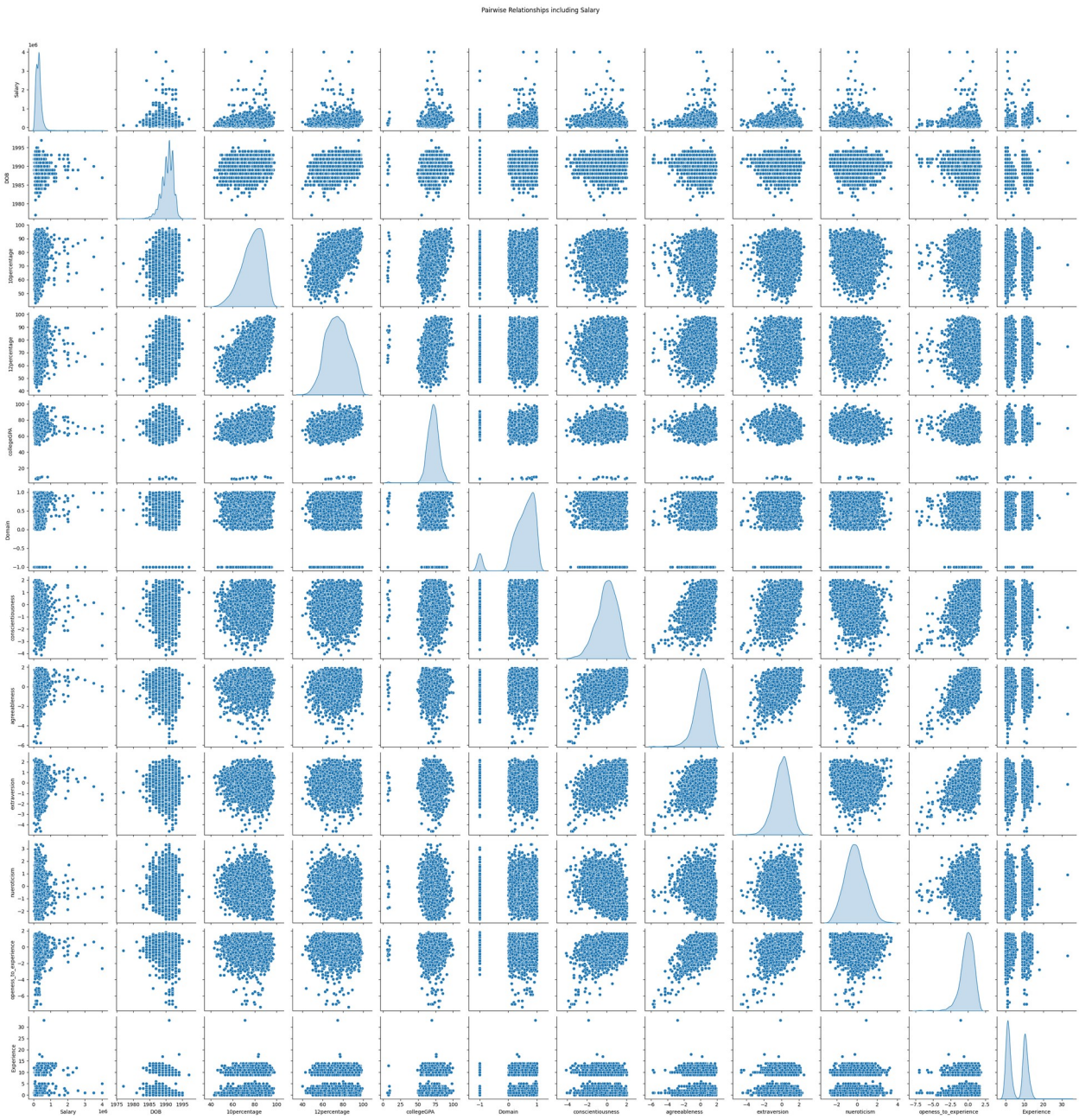



```
num_columns1 = df1.select_dtypes(include=['float64','int32']).columns
# Num_Columns for Numerical Value
num_columns1

Index(['Salary', 'DOB', '10percentage', '12percentage', 'collegeGPA',
'Domain',
       'conscientiousness', 'agreeableness', 'extraversion',
'nueroticism',
       'openess_to_experience', 'Experience'],
      dtype='object')
```

Pair plot to visualize pairwise relationships

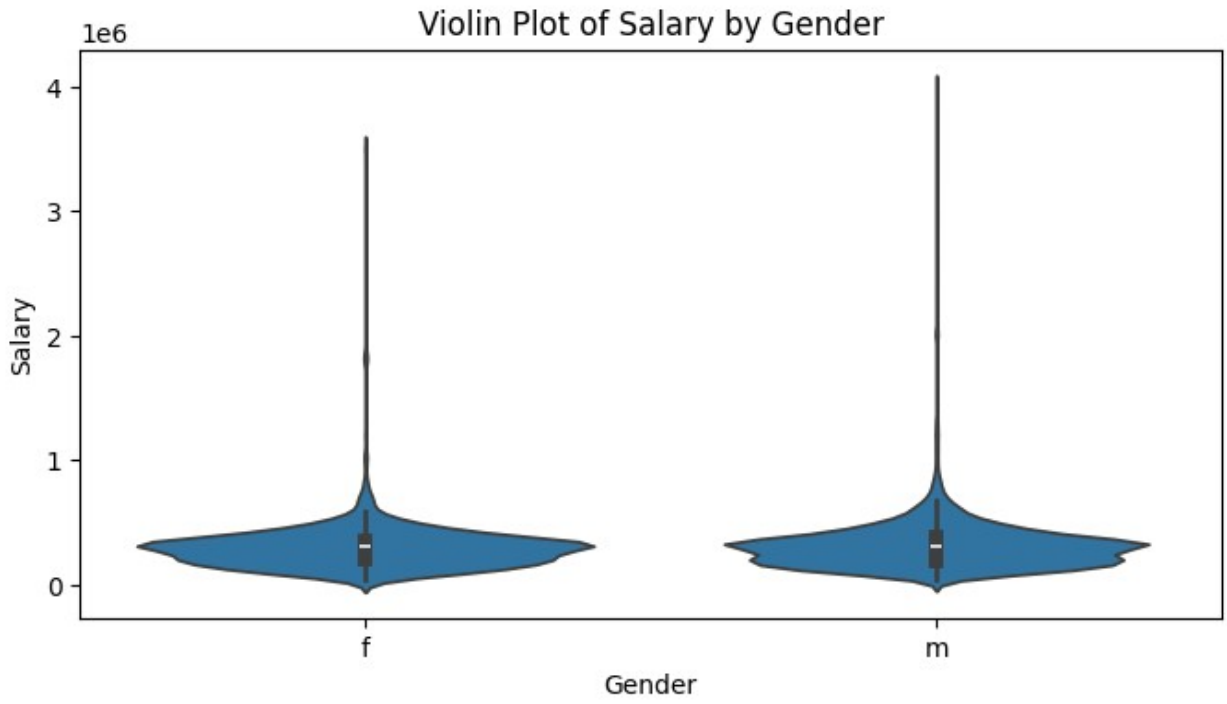
1. pair plot shows the distribution of variable along the diagonal scatter plot of pairwise relationships btw num. variable off diagonal

```
plt.figure(figsize=(10,10))
sns.pairplot(df[num_columns1 ], diag_kind='kde', markers='o')
plt.suptitle('Pairwise Relationships including Salary', y=1.02)
plt.show()

<Figure size 1000x1000 with 0 Axes>
```

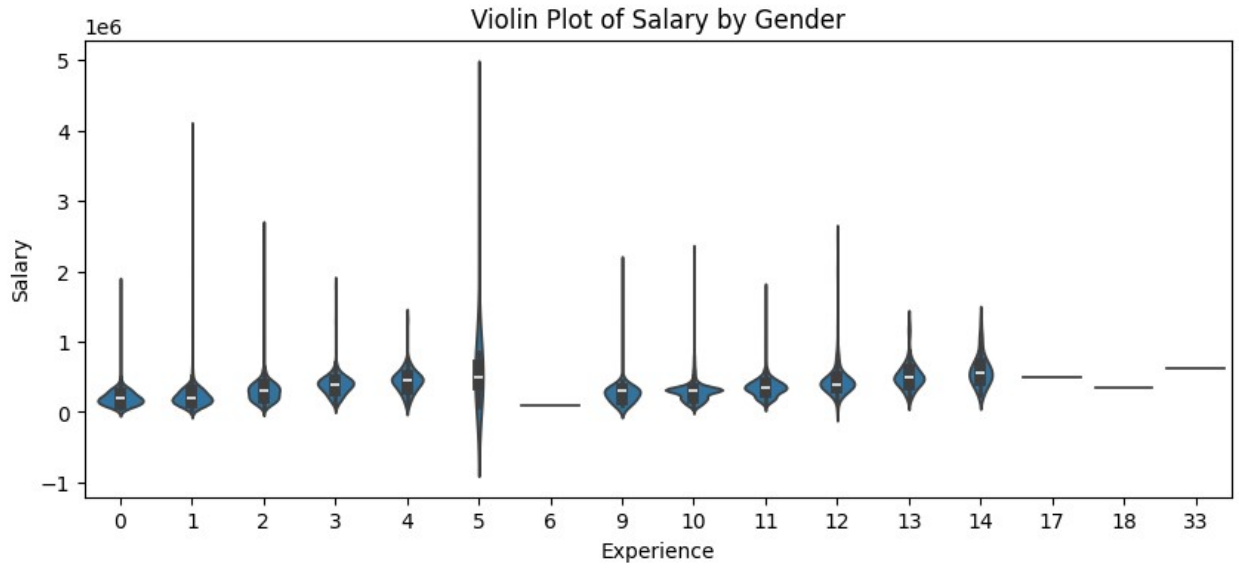



```
plt.figure(figsize=(8, 4))
sns.boxplot(x='Gender', y='Salary', data=df)
plt.title('Box Plot of Gender by Salary')
plt.show()
print("Males (m) tend have higher median salary compared to females ")
```


```
Males (m) tend have higher median salary compared to females

plt.figure(figsize=(8, 4))
sns.violinplot(x='Gender', y='Salary', data=df)
plt.title('Violin Plot of Salary by Gender')
plt.show()
print("violin plot shows the salary distribution for males is wider &
more skewed towards higher values compared to females")
```




```
violin plot shows the salary distribution for males is wider & more
skewed towards higher values compared to females

plt.figure(figsize=(10, 4))
sns.violinplot(x='Experience', y='Salary', data=df)
plt.title('Violin Plot of Salary by Gender')
plt.show()
print("the violin plot shows the distribution of salary across
different levels of experience, plot indicate more candidates  salary
range & overall distribution provides insight into salary variability
across experience levels .  ")
```



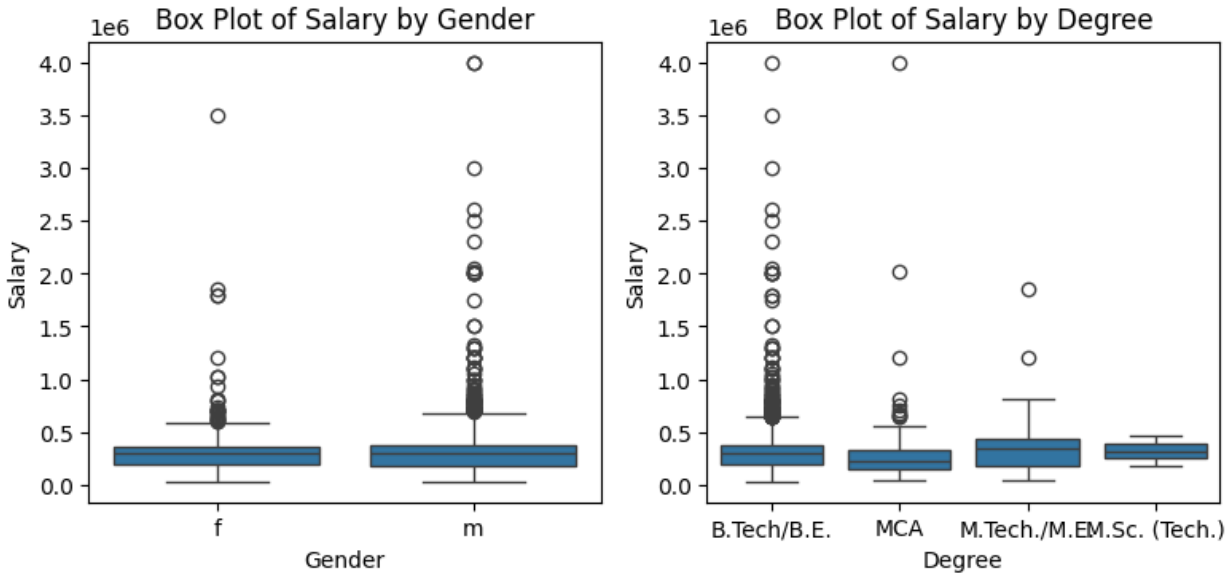
the violin plot shows the distribution of salary across different levels of experience, plot indicate more candidates salary range & overall distribution provides insight into salary variability across experience levels .

```
df1.columns
```

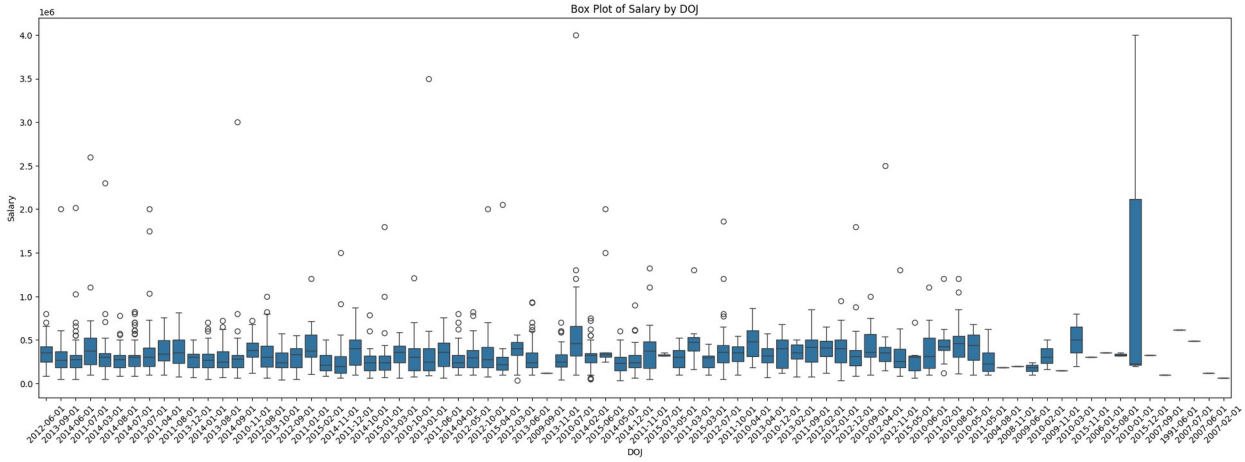
```
Index(['Unnamed: 0', 'ID', 'Salary', 'DOJ', 'DOL', 'Designation',
'JobCity',
       'Gender', 'DOB', '10percentage', '10board', '12graduation',
       '12percentage', '12board', 'CollegeID', 'CollegeTier',
'Degree',
       'Specialization', 'collegeGPA', 'CollegeCityID',
'CollegeCityTier',
     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

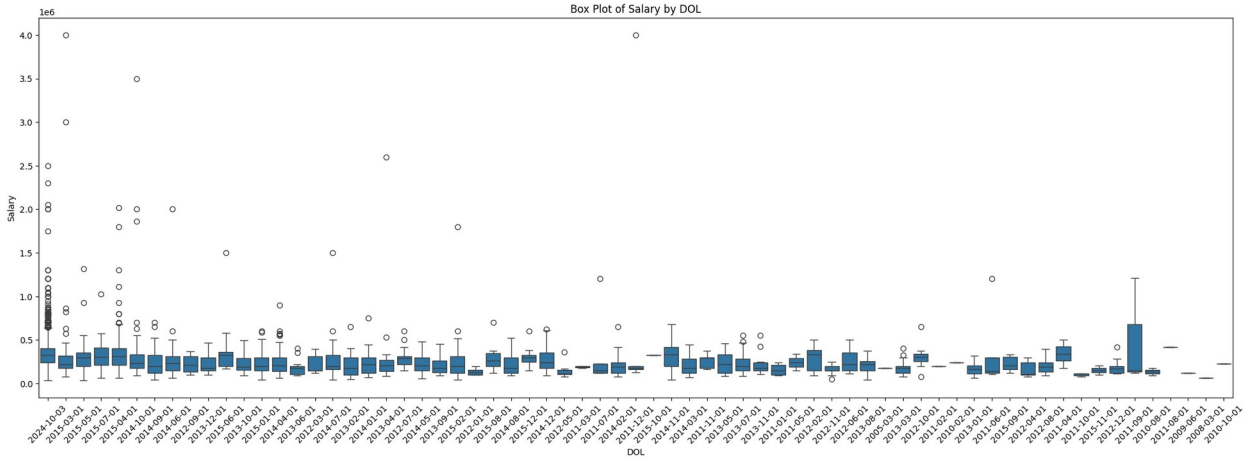
```
categorical_columns = ['Gender','Degree']
# Create a 1x2 subplot layout (1 row, 2 columns)
fig, axs = plt.subplots(1, 2, figsize=(8, 4))
# Iterate through categorical columns and create boxplots
for i, column in enumerate(categorical_columns):
  sns.boxplot(x=df[column], y=df['Salary'], ax=axs[i])  # Use 'Salary'
for y-axis
  axs[i].set_title(f'Box Plot of Salary by {column}')
```

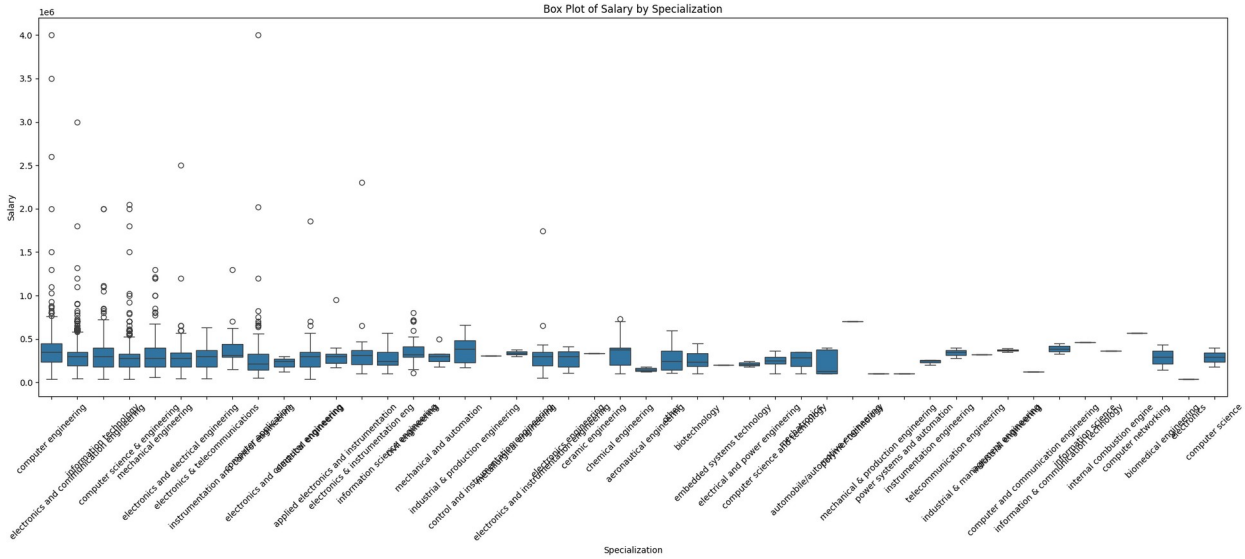
```
  axs[i].set_xlabel(column)
  axs[i].set_ylabel('Salary')
plt.tight_layout() # Adjust spacing between subplots
plt.show()
```

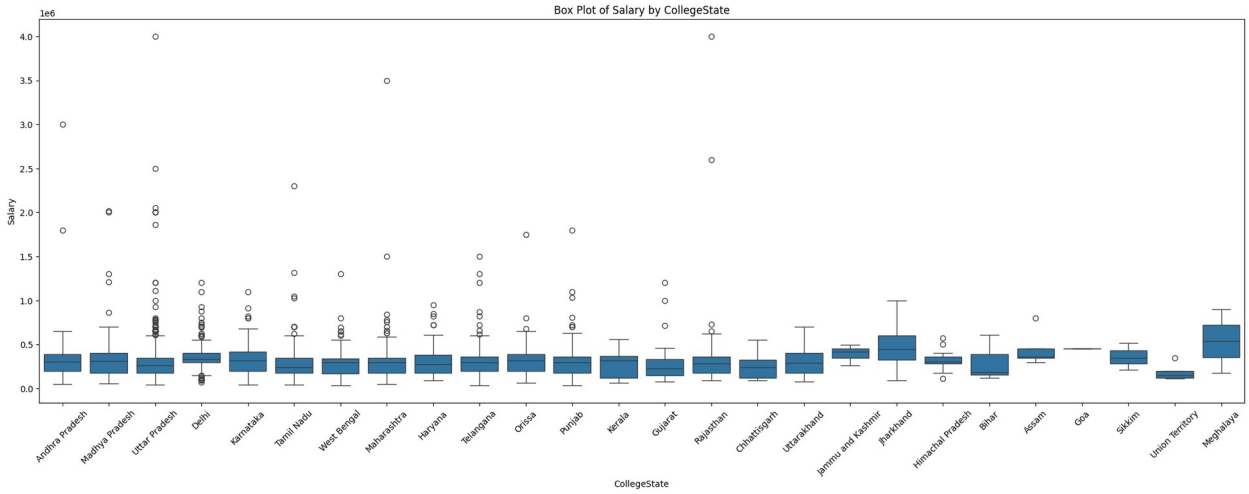



```
categorical_columns = [
    'DOJ', 'DOL', 'Specialization', 'CollegeState'
]
for i, column in enumerate(categorical_columns):
    plt.figure(figsize=(25, 8))
    sns.boxplot(x=df[column], y=df['Salary'])  # Use 'Salary' for y-
axis
    plt.title(f'Box Plot of Salary by {column}')
    plt.xlabel(column)
    plt.ylabel('Salary')
    plt.xticks(rotation=45)
plt.show()
print("box plots show distribution of salary across different cat.
columns (Median salary, range, potential outliers for category are
clearly visible, based on Specialization, collegestate, ")
```

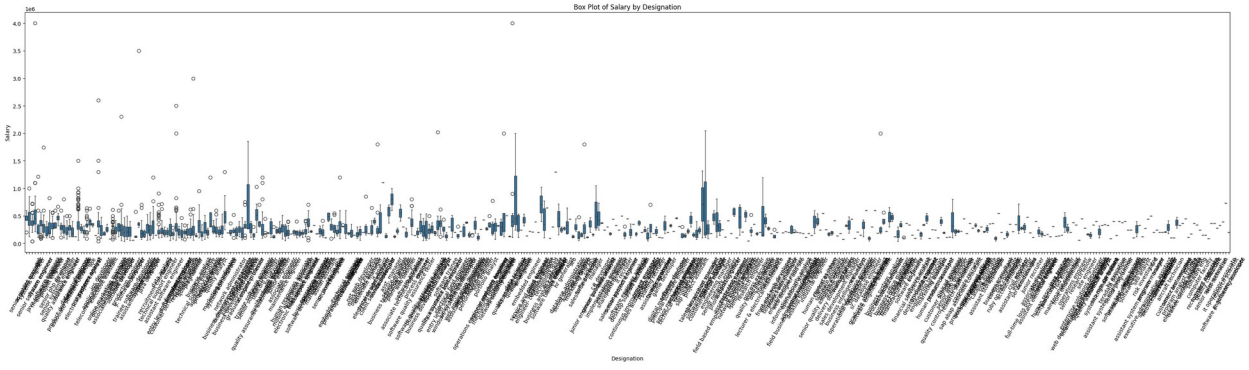

Box Plot of Salary by DOJ
1e6
Salary
DOJ

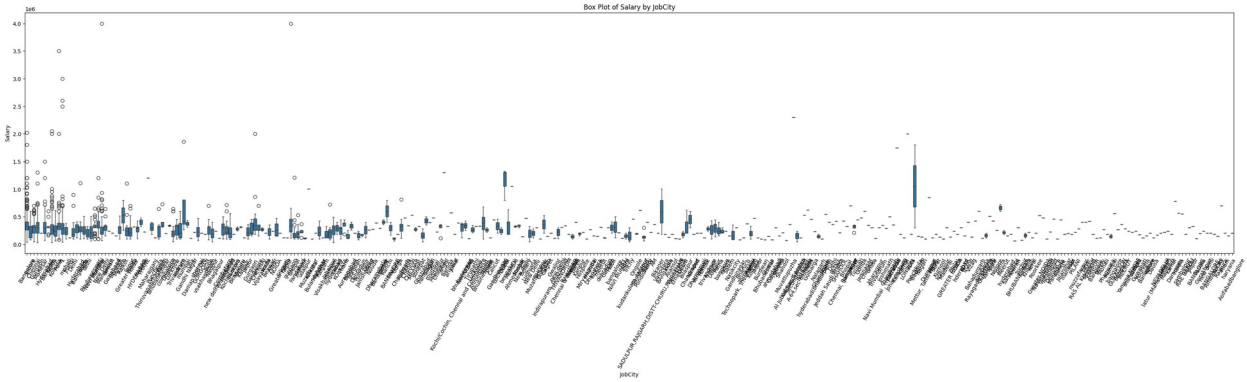

Box Plot of Salary by DOL
1e6
Salary
DOL

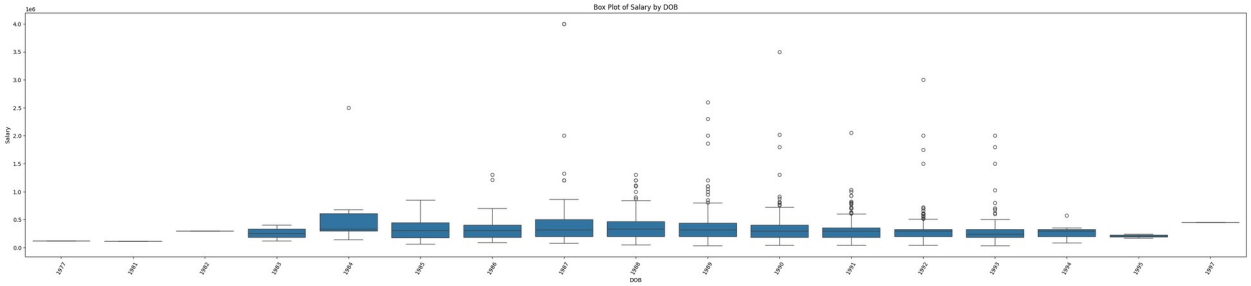

Box Plot of Salary by Specialization
1e6
Salary
Specialization

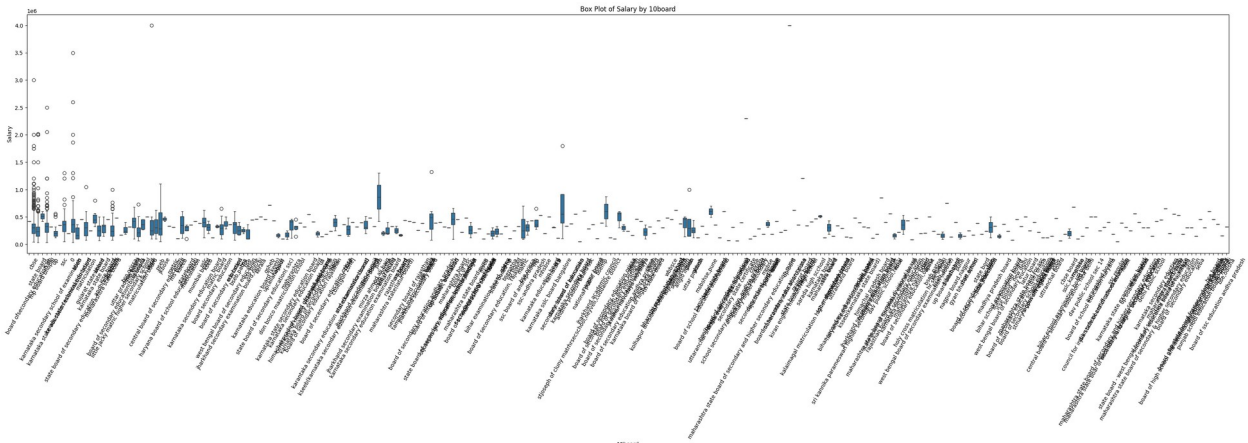


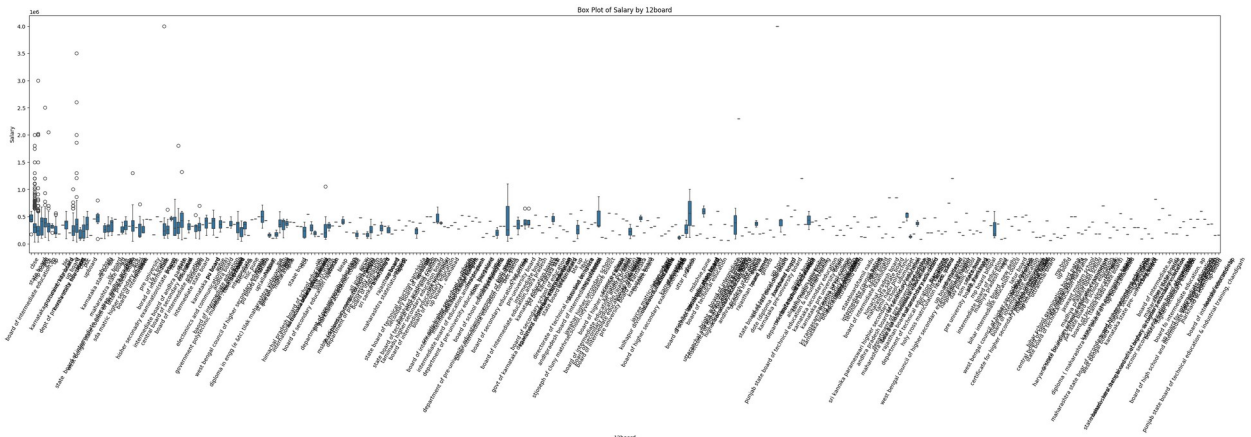

```
box plots show distribution of salary across different cat. columns
(Median salary, range, potential outliers for category are clearly
visible, based on Specialization, collegestate,

categorical_columns = ['Designation', 'JobCity',
    'DOB', '10board', '12board','Experience']
for i, column in enumerate(categorical_columns):
    plt.figure(figsize=(40, 8))
    sns.boxplot(x=df[column], y=df['Salary'])  # Use 'Salary' for y-
axis
    plt.title(f'Box Plot of Salary by {column}')
    plt.xlabel(column)
    plt.ylabel('Salary')
    plt.xticks(rotation=60)
plt.show()
```

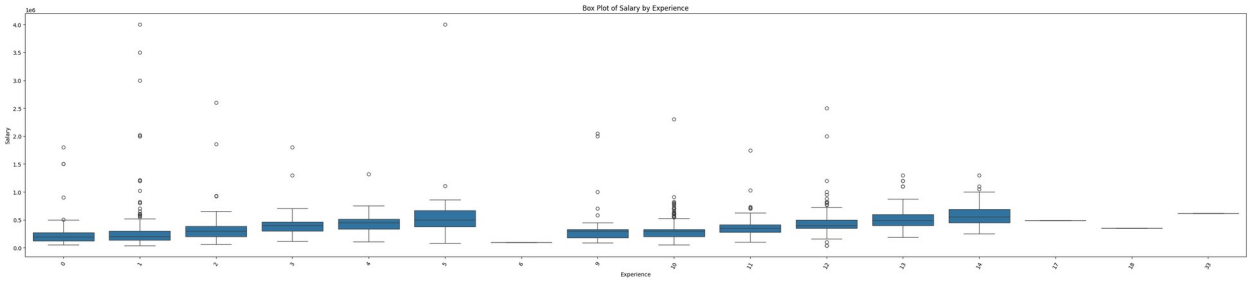



Box Plot of Salary by JobCity
Salary
JobCity


Box Plot of Salary by DOB
Salary
DOB


Box Plot of Salary by 10board
Salary
10board


Box Plot of Salary by 12board
Salary
12board

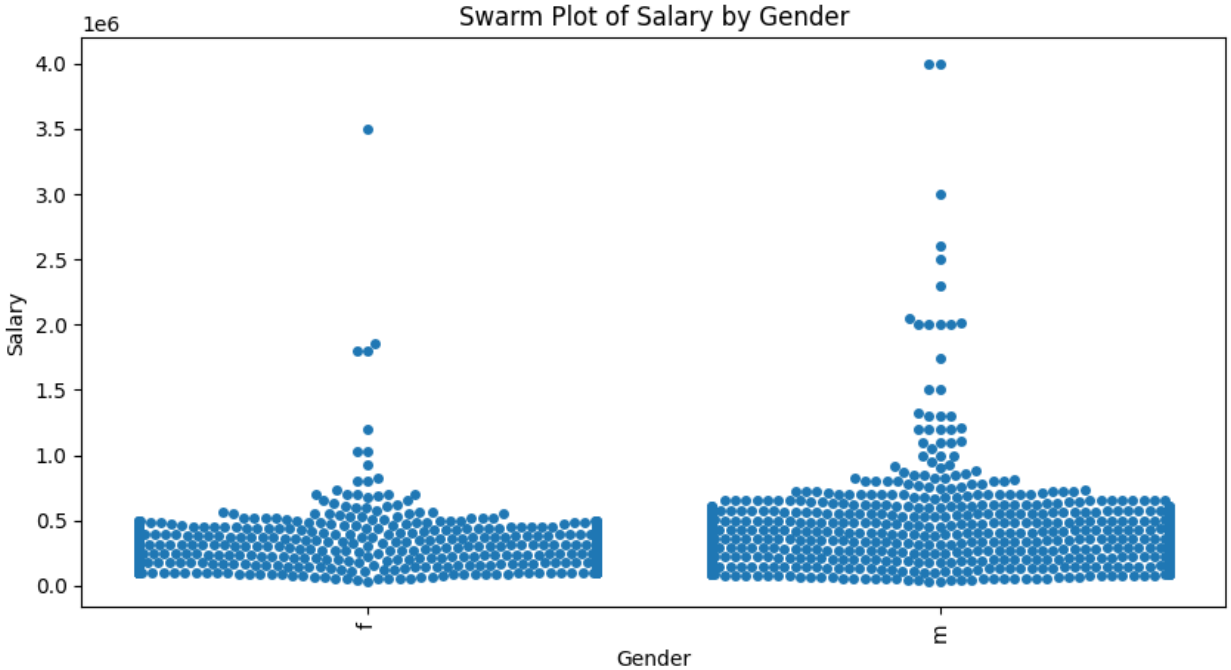


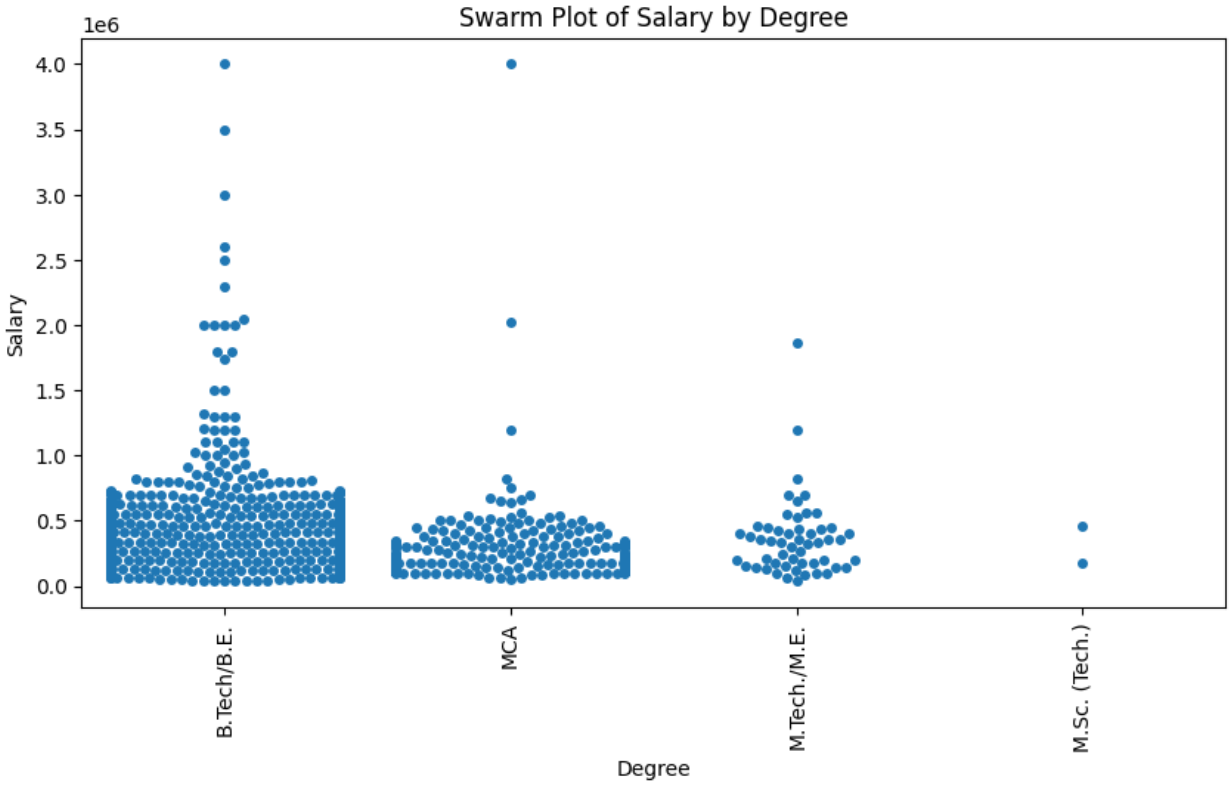

```
categorical_columns = ['Gender','Degree','JobCity',
    'DOB', '10board', '12board','Specialization','Experience']
```

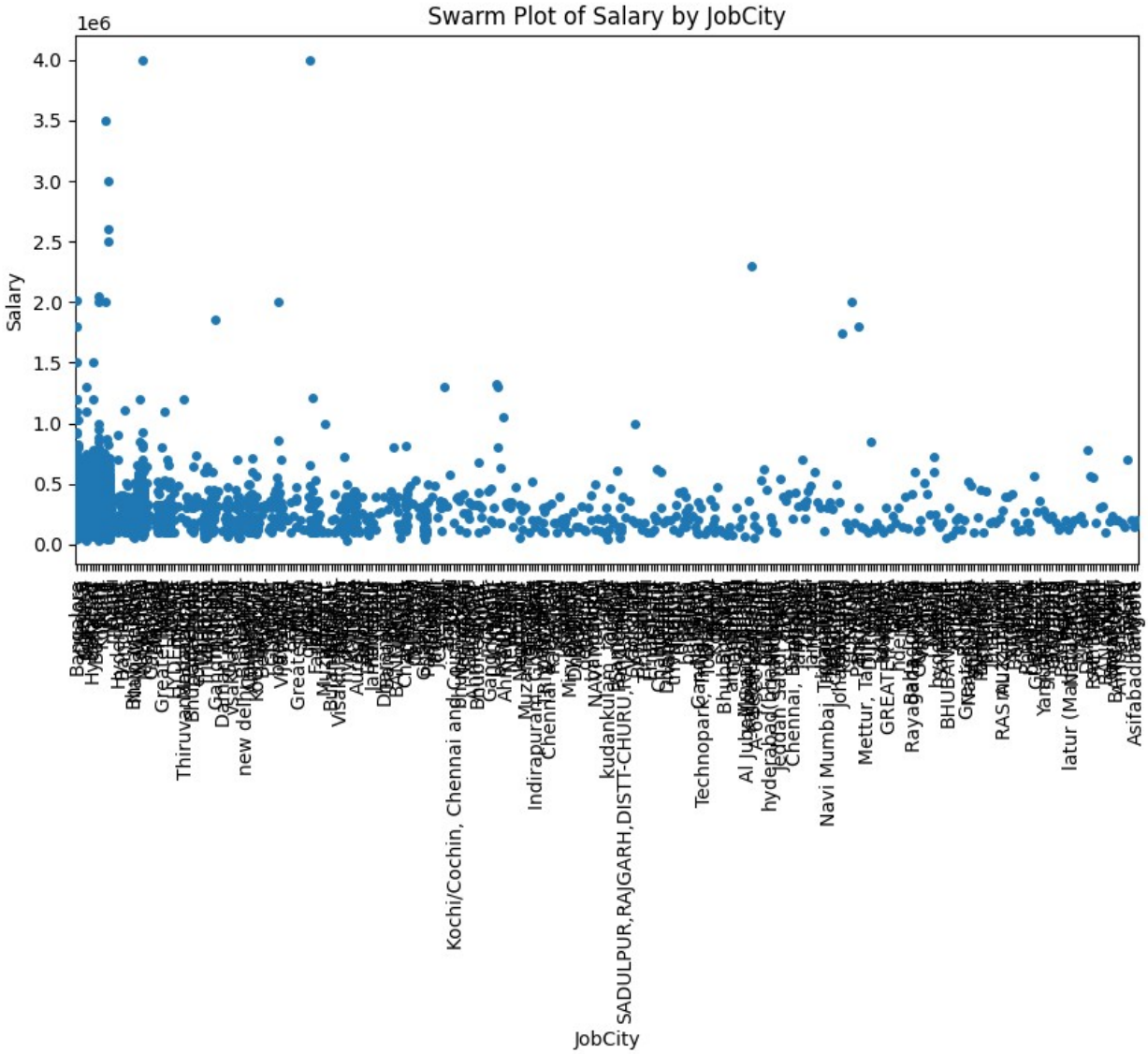
Swarm plots

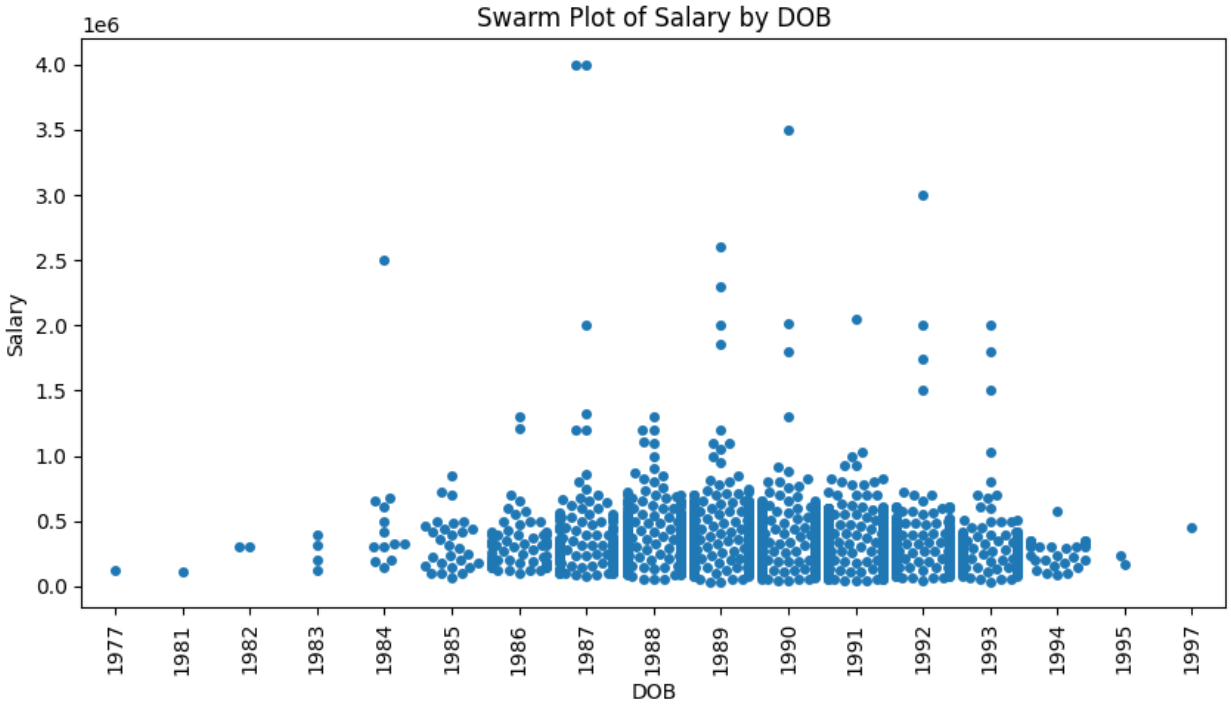
1. Distribution & density of indivdual salary data point across different cat variables visible clustering in categories.
2. Categories like specialization & other cat. columns clear groupings distinct clusters

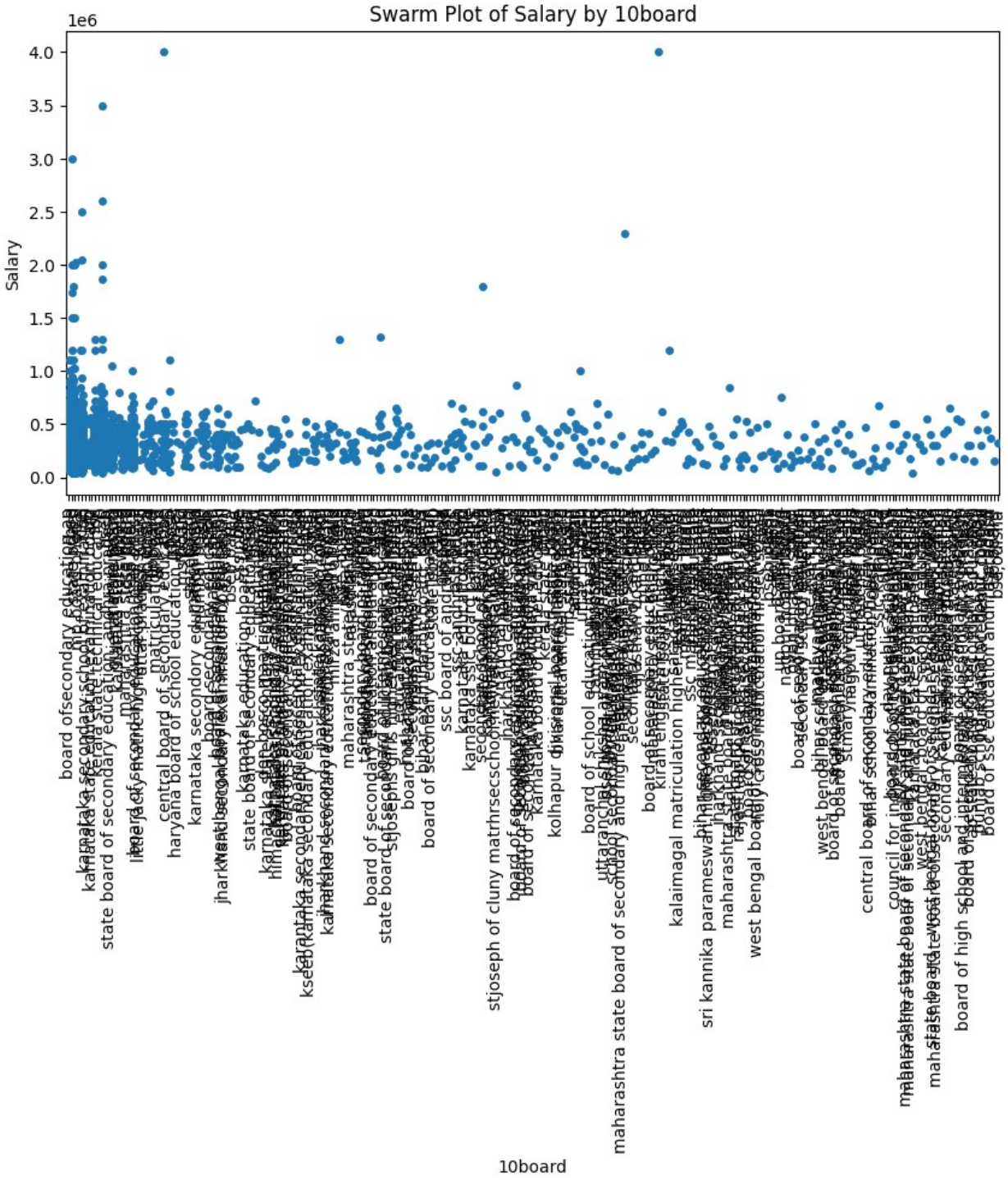
```
# Create swarm plots for Salary based on categorical variables
for column in categorical_columns:
    plt.figure(figsize=(10, 5))
    sns.swarmplot(x=df[column], y=df['Salary'])
    plt.title(f'Swarm Plot of Salary by {column}')
    plt.xlabel(column)
    plt.ylabel('Salary')
    plt.xticks(rotation=90)
    plt.show()
```

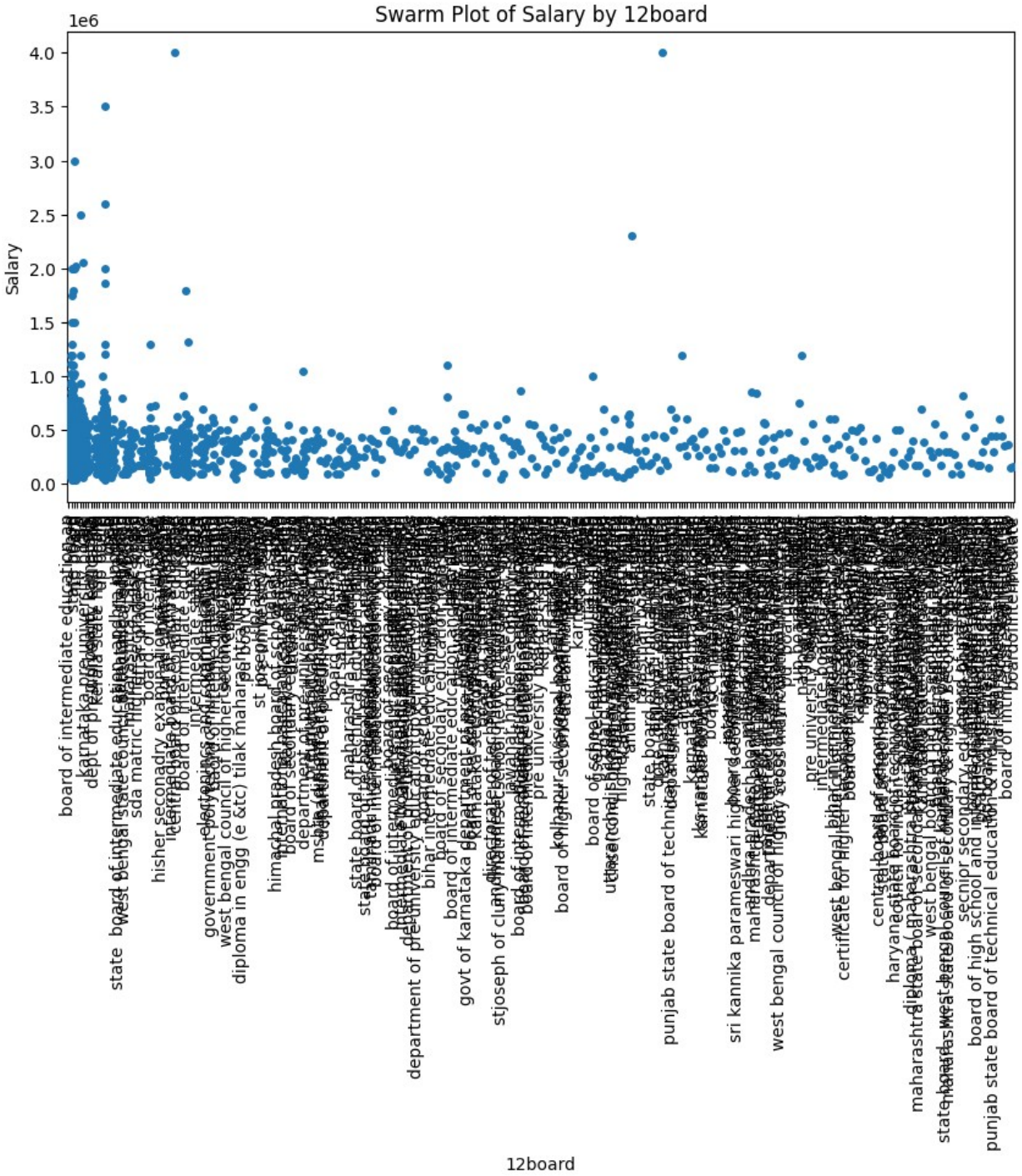


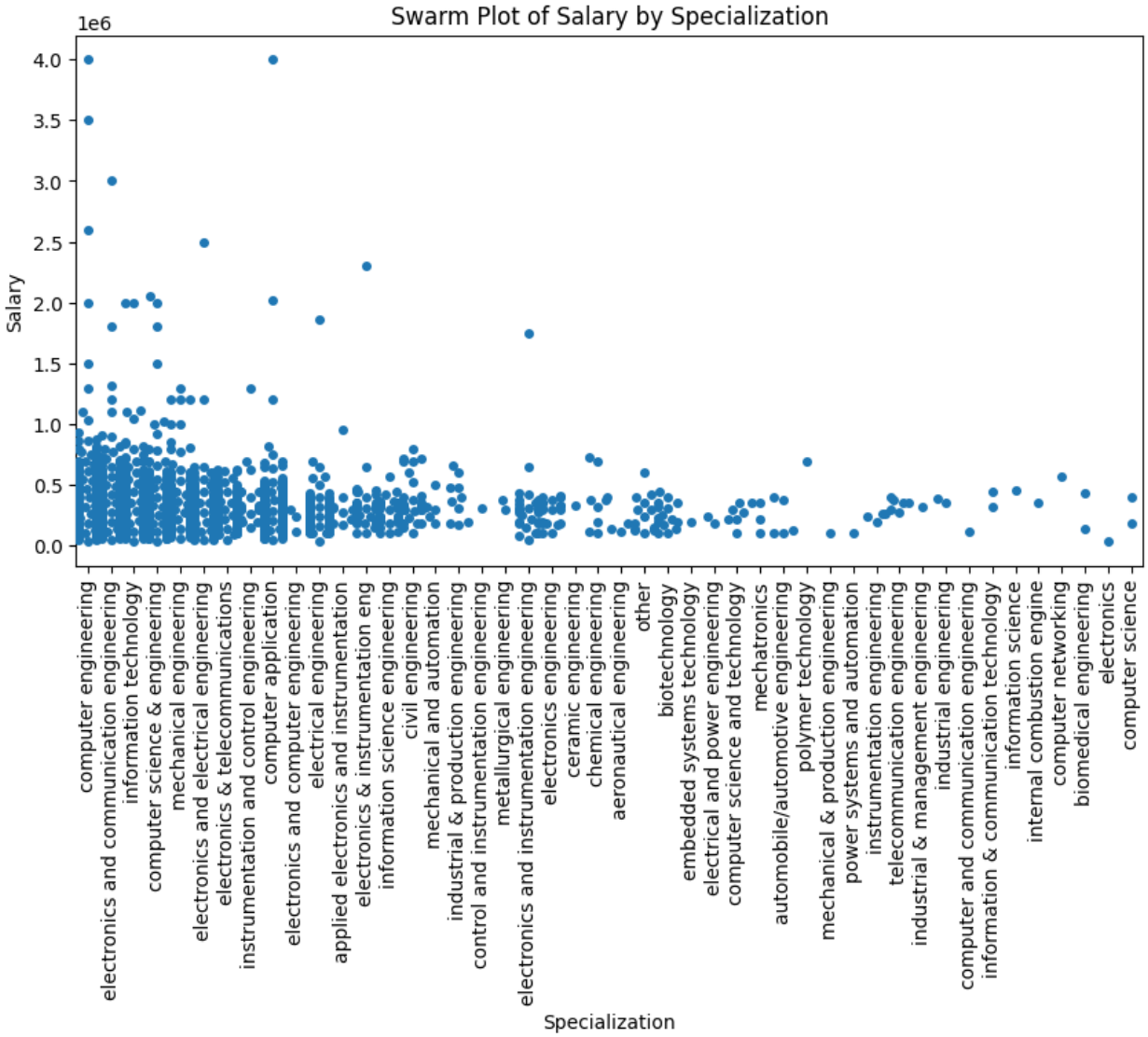

Swarm Plot of Salary by Degree
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)
Degree


Swarm Plot of Salary by JobCity
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
JobCity


Swarm Plot of Salary by DOB
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
1977
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1997
DOB


Swarm Plot of Salary by 10board
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
10board

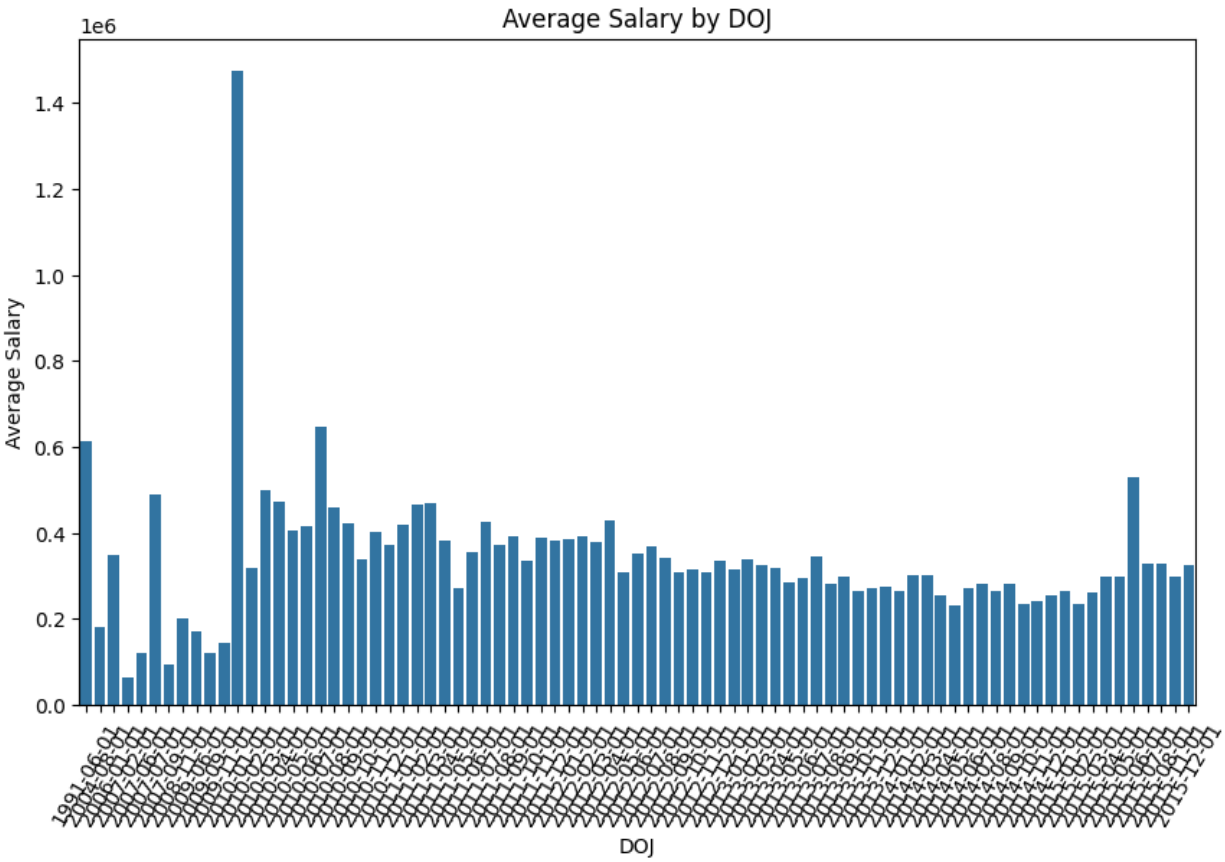

Swarm Plot of Salary by 12board
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
12board

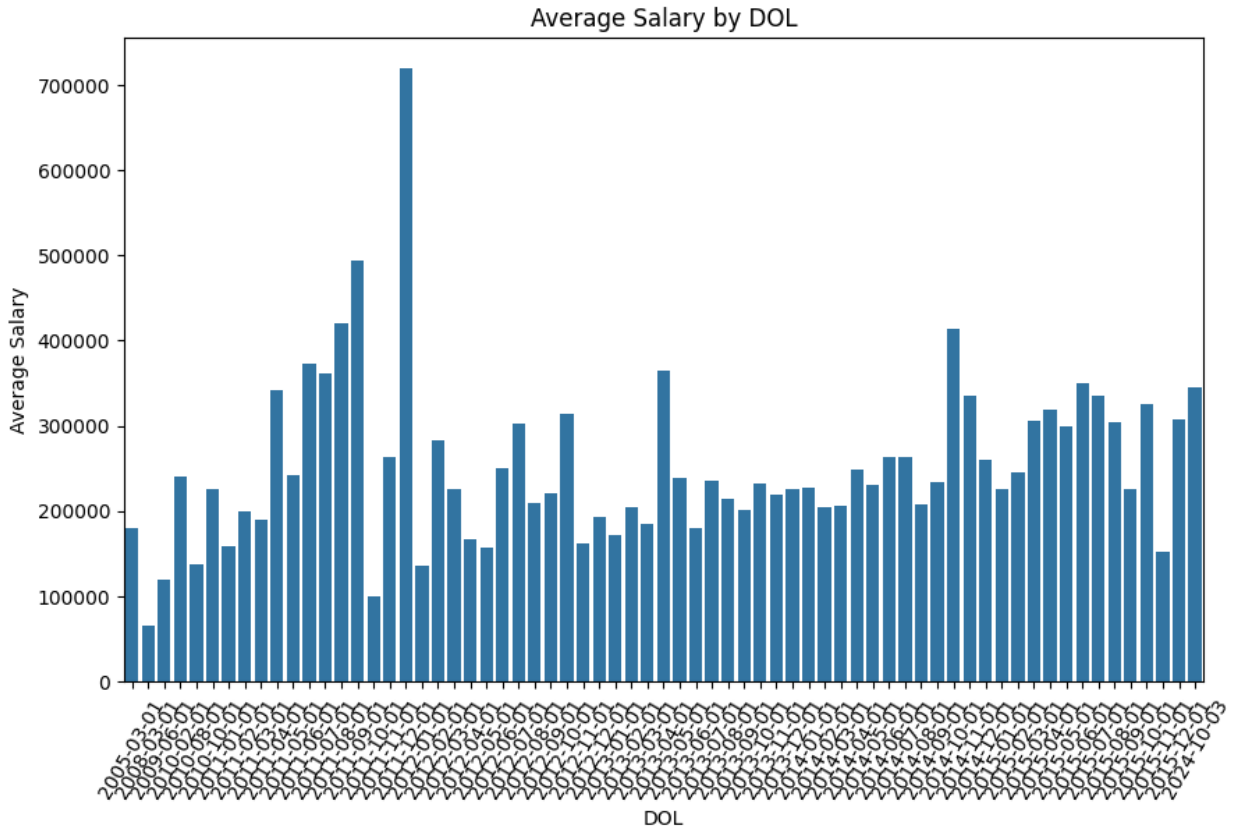

Swarm Plot of Salary by Specialization
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
computer engineering
electronics and communication engineering
information technology
computer science & engineering
mechanical engineering
electronics and electrical engineering
electronics & telecommunications
instrumentation and control engineering
computer application
electronics and computer engineering
electrical engineering
applied electronics and instrumentation
electronics & instrumentation eng
information science engineering
civil engineering
mechanical and automation
industrial & production engineering
control and instrumentation engineering
metallurgical engineering
electronics and instrumentation engineering
electronics engineering
ceramic engineering
chemical engineering
aeronautical engineering
other
biotechnology
embedded systems technology
electrical and power engineering
computer science and technology
mechatronics
automobile/automotive engineering
polymer technology
mechanical & production engineering
power systems and automation
instrumentation engineering
telecommunication engineering
industrial & management engineering
industrial engineering
computer and communication engineering
information & communication technology
information science
internal combustion engine
computer networking
biomedical engineering
electronics
computer science
Specialization

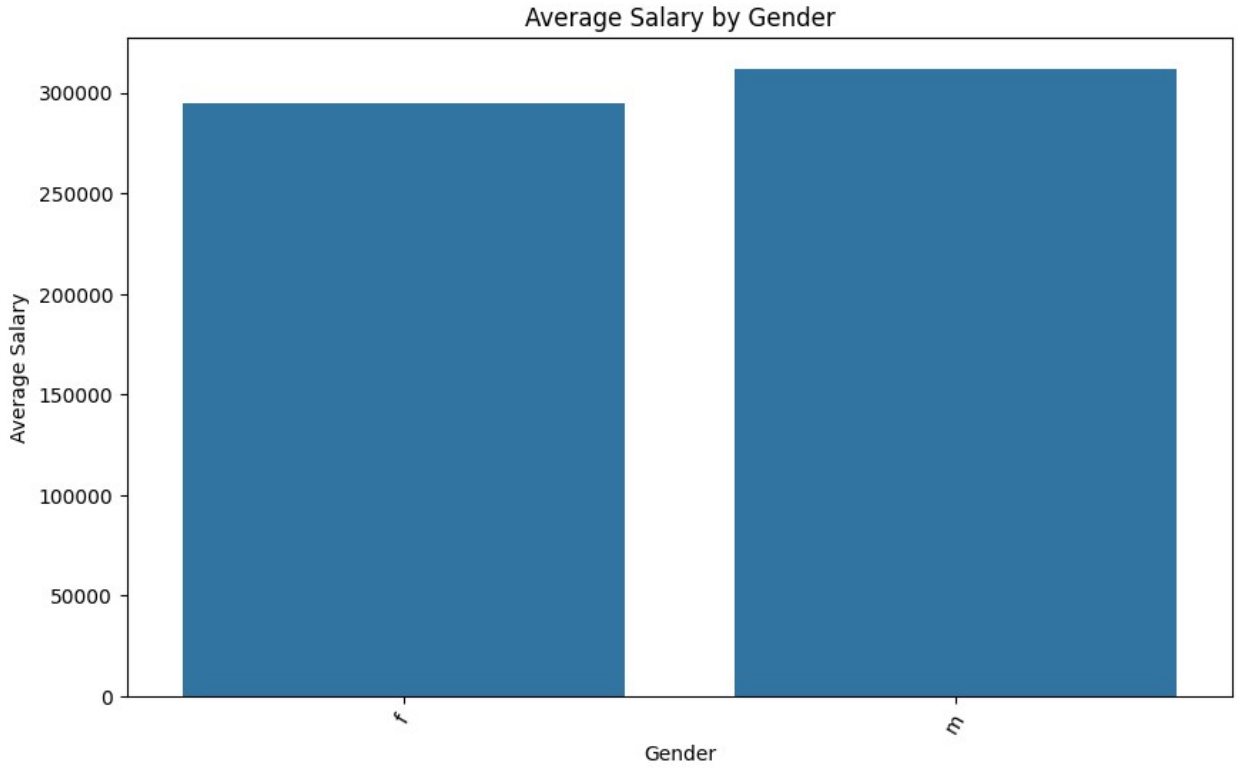

```
print(cat_col)
```

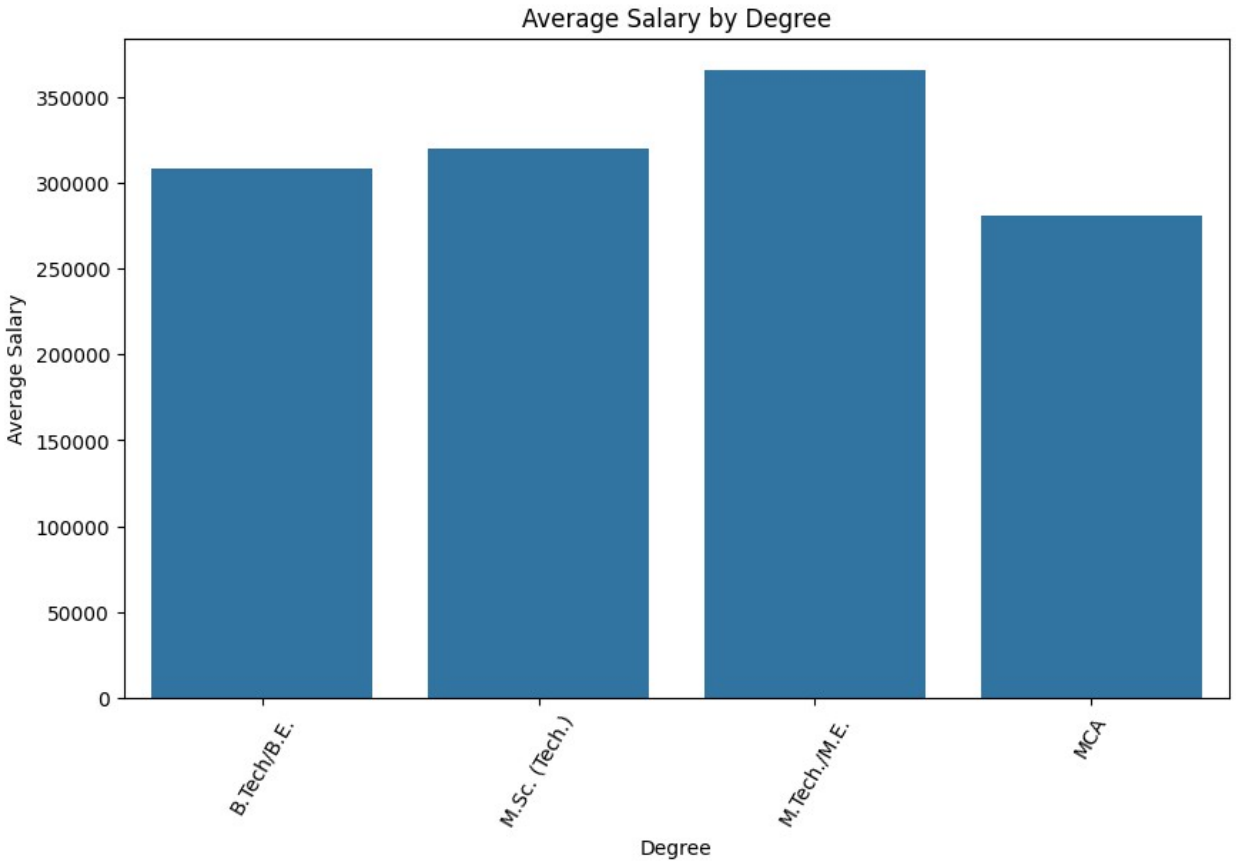
```
Index(['Unnamed: 0', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender',
       '10board', '12board', 'Degree', 'Specialization',
'CollegeState'],
      dtype='object')
```

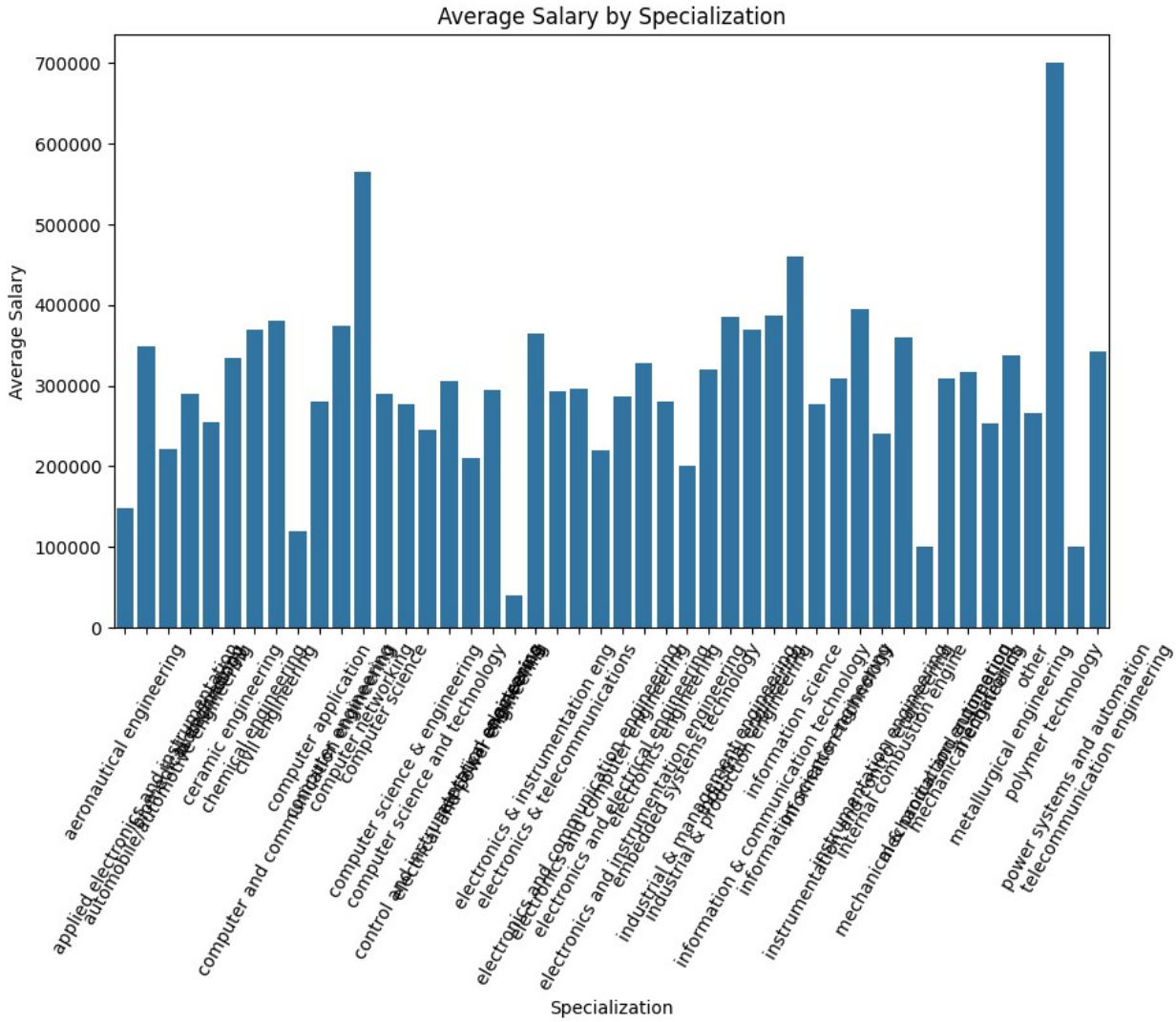
```
# Create bar plots for average Salary by category
cat_col =
['DOJ','DOL','Gender','Degree','Specialization','CollegeState']
for column in cat_col:
    plt.figure(figsize=(10, 6))
    mean_salary = df.groupby(column)['Salary'].mean().reset_index()
    sns.barplot(x=mean_salary[column], y=mean_salary['Salary'])
    plt.title(f'Average Salary by {column}')
    plt.xlabel(column)
    plt.ylabel('Average Salary')
    plt.xticks(rotation=60)
    plt.show()
```

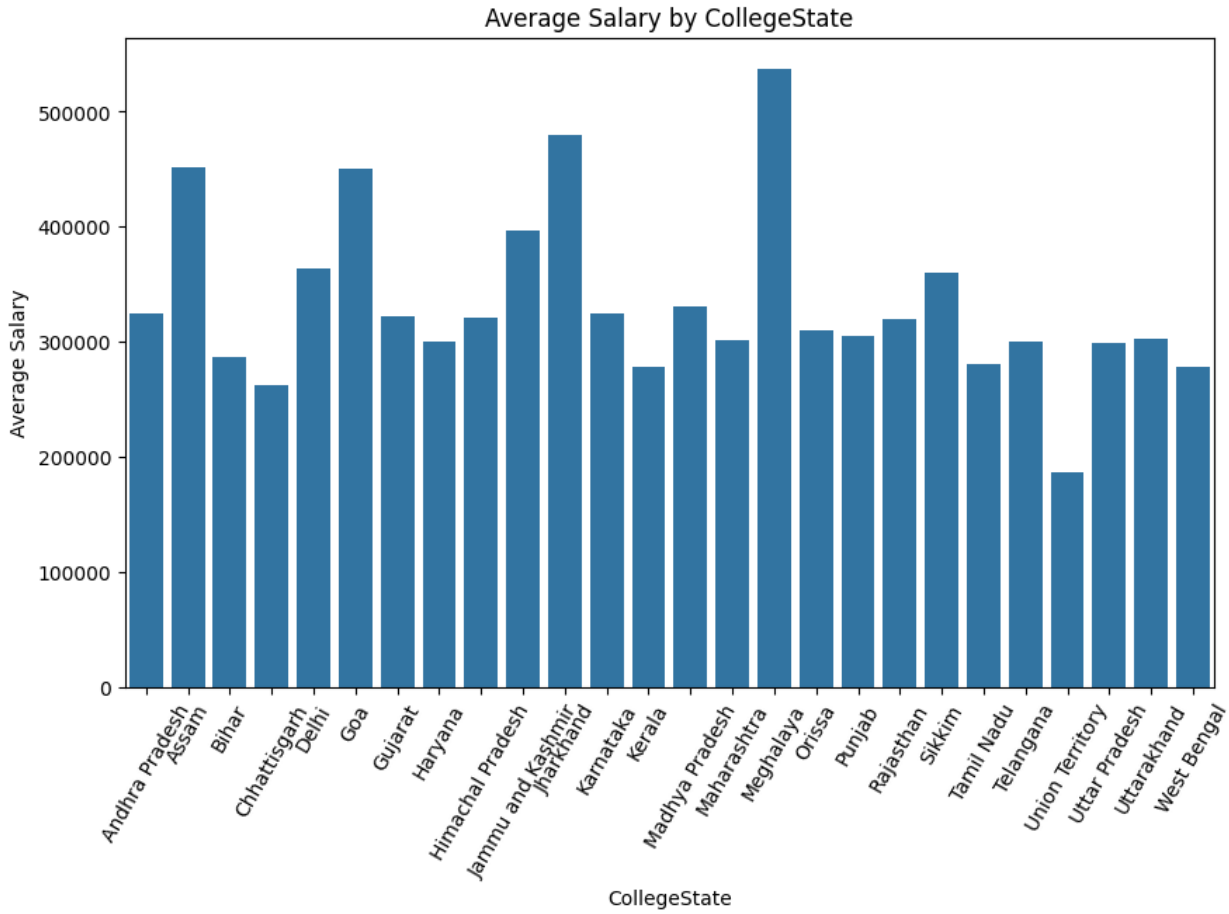

Average Salary by DOJ
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Average Salary
DOJ


Average Salary by DOL
Average Salary
0
100000
200000
300000
400000
500000
600000
700000
DOL


Average Salary by Gender
Average Salary
0
50000
100000
150000
200000
250000
300000
f
m
Gender


Average Salary by Degree
350000
300000
250000
200000
150000
100000
50000
0
Average Salary
B.Tech/B.E.
M.Sc. (Tech.)
M.Tech./M.E.
MCA
Degree

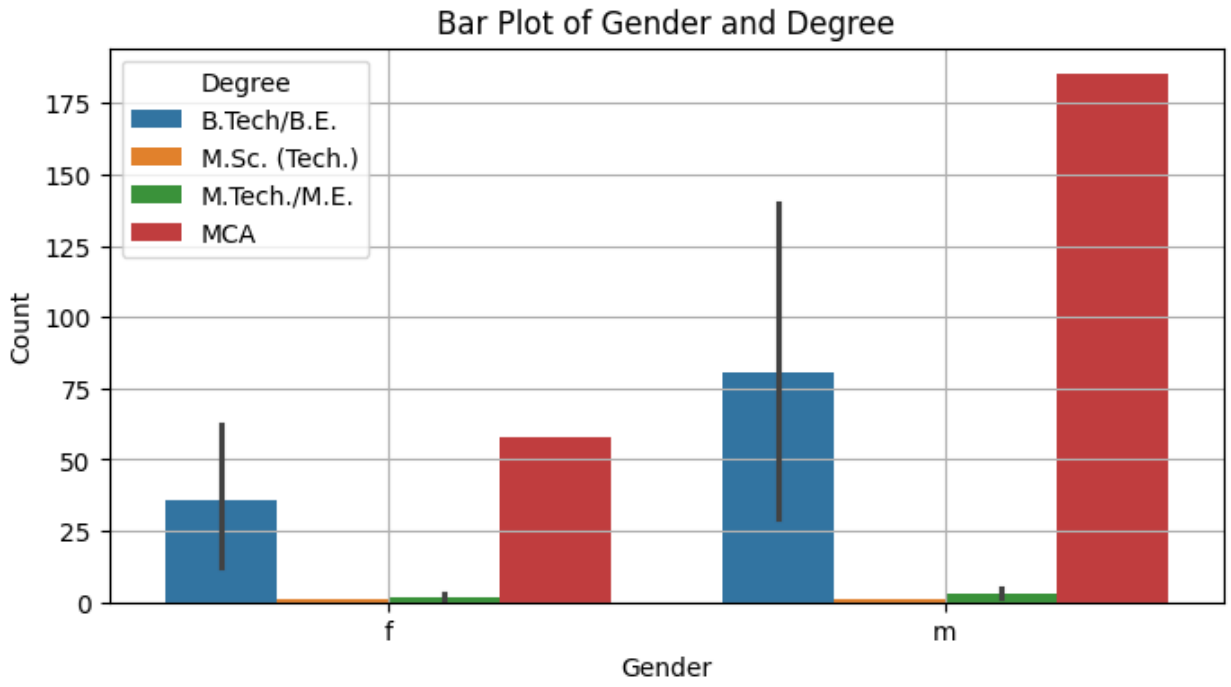

Average Salary by Specialization
Average Salary
0
100000
200000
300000
400000
500000
600000
700000
aeronautical engineering
computer application
computer science
computer science & engineering
computer science and technology
electronics & instrumentation eng
electronics & telecommunications
electronics and computer engineering
electronics and instrumentation engineering
embedded systems technology
industrial & production engineering
information science
information technology
other
metallurgical engineering
polymer technology
power systems and automation
telecommunication engineering
Specialization




```
print('Bar plot distribution of degree among genders indicating
distinct patterns where degrees attract more candidates from one
gender compared to other. ')
```

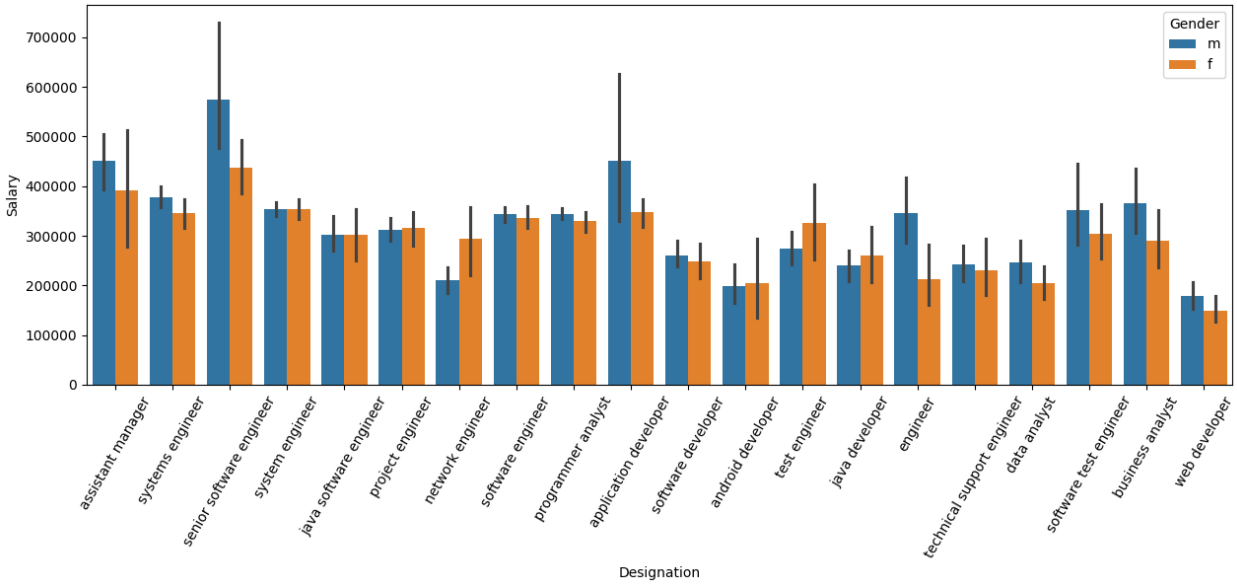
```
Bar plot distribution of degree among genders indicating distinct
patterns where degrees attract more candidates from one gender
compared to other.
```

```
gender_degree_counts = df.groupby(['Gender',
'Degree','Specialization']).size().reset_index(name='Count')
# Create a bar plot
plt.figure(figsize=(8, 4))
sns.barplot(x='Gender', y='Count', hue='Degree',
data=gender_degree_counts)
plt.title('Bar Plot of Gender and Degree')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.grid(True)
plt.show()
```




```
popular_Designation = df['Designation'].value_counts()
[:20].index.tolist()
print(popular_Designation)
top_Designations = df[df['Designation'].isin(popular_Designation)]
# top_designations =
df['Designation'].value_counts().nlargest(10).index.tolist()
plt.figure(figsize=(15,5))
sns.barplot(x='Designation',y='Salary',hue='Gender',data=top_Designati
ons)
plt.xticks(rotation=60)
plt.yticks()
plt.show()
print("Bar plot the average salaries of top 20 designations,
differentiated by gender highlighting that designation tend to have
higher average salaries for males & males. Notable disparities salary
btw genders for specfic designations potential underlying factors
influencing pay equity, warranting further invertigation workplace
practices & policies.")
```

```
['software engineer', 'software developer', 'system engineer',
'programmer analyst', 'systems engineer', 'java software engineer',
'software test engineer', 'project engineer', 'technical support
engineer', 'senior software engineer', 'java developer', 'test
engineer', 'web developer', 'application developer', 'assistant
manager', 'network engineer', 'data analyst', 'business analyst',
'engineer', 'android developer']
```

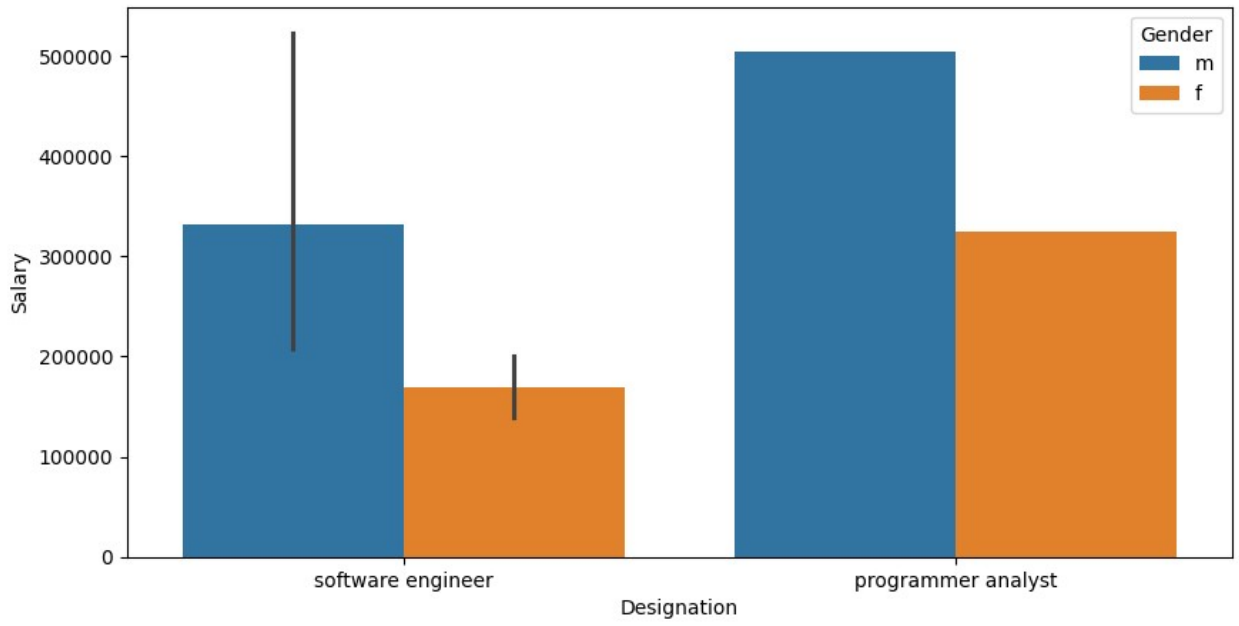



```
Bar plot the average salaries of top 20 designations, differentiated
by gender highlighting that designation tend to have higher average
salaries for males & males. Notable disparities salary btw genders for
specfic designations potential underlying factors influencing pay
equity, warranting further invertigation workplace practices &
policies.
```

Research Questions

- Times of India article dated Jan 18, 2019 states that “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.” Test this claim with the data given to you.

```
new = df[(df["Designation"].isin(["programmer analyst", "software
engineer", "hardware engineer", "associate engineer"])) &
(df['Experience'] == 0)]
plt.figure(figsize=(10, 5))
sns.barplot(x="Designation", y="Salary", hue="Gender", data=new)
plt.show()
```




```
print("Bar plot shows that fresh graduates with 0 years experience in
role of programmer analyst , software Engineer have varying salary
distributions with some exceeding the 2.5-3 lakhs range ")
```

```
Bar plot shows that fresh graduates with 0 years experience in role of
programmer analyst , software Engineer have varying salary
distributions with some exceeding the 2.5-3 lakhs range
```

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

```
from scipy.stats import chi2_contingency
# Create a contingency table for Gender vs. Specialization
gender_specialization_contingency = pd.crosstab(df['Gender'],
df['Specialization'])
# Perform chi-square test to determine if there is a significant
relationship between gender and specialization
chi2_stat, p_val, dof, ex =
chi2_contingency(gender_specialization_contingency)
# Determine the significance level
alpha = 0.05
# Interpret the p-value
if p_val < alpha:
    relationship_result = "There is a significant relationship between
gender and specialization."
else:
    relationship_result = "There is no significant relationship
between gender and specialization."
print("Result of Chi-square Test for Gender vs. Specialization:")
print("Chi-square Statistic:", chi2_stat)
```

```
print("p-value:", p_val)
print("Result:", relationship_result)
```

```
Result of Chi-square Test for Gender vs. Specialization:
Chi-square Statistic: 104.46891913608455
p-value: 1.2453868176976918e-06
Result: There is a significant relationship between gender and
specialization.
```

```
print("CHi-square test results indicate a statistically significant
relationship btw gender & specialization the p-vlaue is below the
alpha level of 0.05 that gender may influence specialization choices")
print('Calculated chi-square statistic reflects the degree of
association warranting a deeper investigation into the factors driving
perferences different genders. ')
```

```
CHi-square test results indicate a statistically significant
relationship btw gender & specialization the p-vlaue is below the
alpha level of 0.05 that gender may influence specialization choices
Calculated chi-square statistic reflects the degree of association
warranting a deeper investigation into the factors driving perferences
different genders.
```

Conclusion

- Based on the Analysis made and we have found the relationship between the target variable Salary and other variables. Here are some insights
- Male has the higher salary compared to female - Senior software Engineer has the highest salary compared to other Designations - Most of the software engineers are from Specilization Computer science , Electronics , Information Technology