amcat-eda

October 4, 2024

1 import Libraries

[1]:
```
import numpy as np
import pandas as pd
```

[2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

2 Read data

[3]:
```
df=pd.read_csv("/content/data.xlsx - Sheet1.csv")
```

[4]:
```
df.sample(10)
```

[4]:

	Unnamed: 0	ID	Salary	DOJ	DOL	\
3767	train	1063305	240000.0	8/1/13 0:00	5/1/15 0:00	
1457	train	1044759	330000.0	8/1/14 0:00	present	
1668	train	540835	700000.0	10/1/12 0:00	present	
478	train	815859	330000.0	8/1/14 0:00	present	
2101	train	358268	450000.0	3/1/13 0:00	present	
448	train	712297	300000.0	8/1/14 0:00	5/1/15 0:00	
3642	train	812555	150000.0	10/1/13 0:00	present	
3848	train	801323	350000.0	6/1/14 0:00	present	
3371	train	1231990	205000.0	8/1/13 0:00	present	
1357	train	330355	500000.0	6/1/12 0:00	present	

	Designation	JobCity	Gender	DOB	\
3767	software developer	Lucknow	m	4/19/90 0:00	
1457	information security analyst	Chennai	m	1/5/92 0:00	
1668	technical support engineer	Bangalore	f	8/1/89 0:00	
478	network engineer	Gurgaon	m	11/6/91 0:00	
2101	system engineer	Mumbai	m	5/6/90 0:00	
448	junior research fellow	Jabalpur	m	8/17/91 0:00	
3642	java software engineer	Banaglore	m	10/11/90 0:00	
3848	software quality assurance tester	Pune	m	2/17/92 0:00	

```
3371                  quality engineer  Hyderabad      f  11/19/91 0:00
1357                production engineer  Hyderabad      m   2/19/91 0:00

      10percentage  … ComputerScience  MechanicalEngg  ElectricalEngg  \
3767          77.0  …               -1              -1              -1
1457          89.6  …               -1              -1              -1
1668          73.0  …               -1              -1              -1
478           74.0  …               -1              -1              -1
2101          76.0  …               -1              -1              -1
448           93.5  …               -1             553              -1
3642          85.0  …               -1              -1              -1
3848          78.6  …              469              -1              -1
3371          85.4  …               -1              -1              -1
1357          84.0  …               -1             383              -1

     TelecomEngg  CivilEngg  conscientiousness agreeableness extraversion  \
3767          -1         -1            -1.7389       -0.4536      -1.6807
1457          -1         -1             0.7027        0.2124      -0.2974
1668          -1         -1             0.9737        0.0328      -0.3440
478           -1         -1             0.9900        0.5454       0.1637
2101          -1         -1            -4.1267       -0.2793      -0.1988
448           -1         -1            -1.3080        0.3789      -0.1437
3642          -1         -1             1.7081        0.2124       2.0080
3848          -1         -1            -0.0154        0.8784       0.9322
3371          -1         -1             0.9900        1.2114       1.3933
1357          -1         -1             0.9737       -0.3183      -0.9245

      nueroticism  openess_to_experience
3767     -1.12180                -1.6273
1457      0.39950                 1.0554
1668     -0.17270                -0.0506
478      -1.24860                -0.6692
2101      1.23740                -0.7615
448      -0.74150                 0.0973
3642     -2.38950                 1.2470
3848      0.39950                 1.2470
3371      2.55460                 0.6721
1357      0.88483                -0.6035

[10 rows x 39 columns]
```

[5]:
```
df.columns
```

[5]:
```
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')
```

[6]:
```
df.info()
```

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

```
 33  CivilEngg              3998 non-null   int64
 34  conscientiousness      3998 non-null   float64
 35  agreeableness          3998 non-null   float64
 36  extraversion           3998 non-null   float64
 37  nueroticism            3998 non-null   float64
 38  openess_to_experience  3998 non-null   float64
dtypes: float64(10), int64(17), object(12)
memory usage: 1.2+ MB
```

[7]:
```
df = df.drop('Unnamed: 0', axis=1)
```

[8]:
```
df.columns
```

[8]:
```
Index(['ID', 'Salary', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DOB',
       '10percentage', '10board', '12graduation', '12percentage', '12board',
       'CollegeID', 'CollegeTier', 'Degree', 'Specialization', 'collegeGPA',
       'CollegeCityID', 'CollegeCityTier', 'CollegeState', 'GraduationYear',
       'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
       'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

[9]:
```
df.head()
```

[9]:

	ID	Salary	DOJ	DOL	Designation	\
0	203097	420000.0	6/1/12 0:00	present	senior quality engineer	
1	579905	500000.0	9/1/13 0:00	present	assistant manager	
2	810601	325000.0	6/1/14 0:00	present	systems engineer	
3	267447	1100000.0	7/1/11 0:00	present	senior software engineer	
4	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	

	JobCity	Gender	DOB	10percentage	\
0	Bangalore	f	2/19/90 0:00	84.3	
1	Indore	m	10/4/89 0:00	85.4	
2	Chennai	f	8/3/92 0:00	85.0	
3	Gurgaon	m	12/5/89 0:00	85.6	
4	Manesar	m	2/27/91 0:00	78.0	

	10board	…	ComputerScience	MechanicalEngg	\
0	board ofsecondary education,ap	…	-1	-1	
1	cbse	…	-1	-1	
2	cbse	…	-1	-1	
3	cbse	…	-1	-1	
4	cbse	…	-1	-1	

```
  ElectricalEngg  TelecomEngg  CivilEngg conscientiousness agreeableness  \
0             -1           -1         -1            0.9737        0.8128
1             -1           -1         -1           -0.7335        0.3789
2             -1           -1         -1            0.2718        1.7109
3             -1           -1         -1            0.0464        0.3448
4             -1           -1         -1           -0.8810       -0.2793

   extraversion  nueroticism  openess_to_experience
0        0.5269      1.35490                -0.4455
1        1.2396     -0.10760                 0.8637
2        0.1637     -0.86820                 0.6721
3       -0.3440     -0.40780                -0.9194
4       -1.0697      0.09163                -0.1295

[5 rows x 38 columns]
```

[10]:
```
df.describe()
```

[10]:
```
                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]
```

3 Data Cleaning

4 checking noise in the given Data

[11]:
```
df['ID'].unique()
```

[11]:
```
array([203097, 579905, 810601, …, 355888, 947111, 324966])
```

[12]:
```
df['Salary'].unique()
```

[12]:
```
array([ 420000.,  500000.,  325000., 1100000.,  200000.,  300000.,
        400000.,  600000.,  230000.,  450000.,  270000.,  350000.,
        250000.,  120000.,  320000.,  190000.,  180000.,  335000.,
        435000.,  345000.,  145000.,  220000.,  530000.,  340000.,
        360000.,  215000.,   80000.,  330000.,  380000.,  110000.,
        205000.,   95000.,  390000.,   60000.,  240000.,  525000.,
        305000.,  150000.,  310000.,  455000.,  800000.,  100000.,
        280000.,  445000.,  315000.,  370000.,  275000., 1500000.,
        425000.,  470000.,  460000.,  510000.,  480000.,  170000.,
        640000.,  225000.,  440000., 1200000.,  675000.,  105000.,
        195000.,  385000.,  235000.,  615000.,  290000.,  140000.,
        405000., 1860000.,  375000.,  430000.,  660000.,   70000.,
        410000.,  550000.,   35000.,  115000.,  415000.,  265000.,
```

```
 285000.,  245000.,  395000.,  560000.,  700000.,  185000.,
 160000.,  625000.,   85000.,  135000.,  785000.,  210000.,
 155000.,  355000.,  535000.,  690000.,  260000., 1110000.,
1000000.,  505000.,  475000.,  715000.,  820000.,   90000.,
 720000., 2600000.,  515000.,   55000.,  495000.,   65000.,
 655000.,  545000.,  520000.,  645000., 1025000.,  775000.,
 490000., 1300000., 3500000.,  910000.,  570000.,  255000.,
 130000.,  175000.,  730000.,  555000.,  465000.,  680000.,
 165000.,  630000.,  365000., 1050000., 2000000.,  860000.,
 125000.,   50000.,  580000.,  485000., 4000000., 2020000.,
 650000.,   45000.,  610000.,  760000.,  585000.,  620000.,
 870000., 2050000.,  540000.,  144000.,  605000., 1320000.,
 755000.,  880000., 3000000.,   75000.,  295000.,   40000.,
 575000.,  565000., 2500000., 2300000.,  590000.,  950000.,
1800000.,  725000.,  930000.,  750000.,  705000., 1745000.,
 850000.,  845000.,  670000., 1030000.,  770000.,  900000.,
1210000.,  810000.,  925000.])
```

[13]:
```
df['DOJ'].unique()
```

[13]:
```
array(['6/1/12 0:00', '9/1/13 0:00', '6/1/14 0:00', '7/1/11 0:00',
       '3/1/14 0:00', '8/1/14 0:00', '7/1/14 0:00', '7/1/13 0:00',
       '4/1/11 0:00', '8/1/11 0:00', '12/1/13 0:00', '1/1/14 0:00',
       '8/1/13 0:00', '9/1/14 0:00', '11/1/10 0:00', '8/1/12 0:00',
       '10/1/13 0:00', '9/1/12 0:00', '1/1/11 0:00', '2/1/15 0:00',
       '11/1/14 0:00', '12/1/11 0:00', '10/1/14 0:00', '1/1/15 0:00',
       '3/1/13 0:00', '10/1/10 0:00', '1/1/13 0:00', '6/1/11 0:00',
       '4/1/14 0:00', '5/1/12 0:00', '10/1/12 0:00', '4/1/15 0:00',
       '3/1/12 0:00', '6/1/13 0:00', '9/1/09 0:00', '11/1/13 0:00',
       '7/1/10 0:00', '2/1/14 0:00', '6/1/15 0:00', '5/1/14 0:00',
       '12/1/14 0:00', '11/1/11 0:00', '7/1/15 0:00', '5/1/13 0:00',
       '3/1/11 0:00', '3/1/15 0:00', '7/1/12 0:00', '10/1/11 0:00',
       '4/1/10 0:00', '4/1/13 0:00', '12/1/10 0:00', '2/1/13 0:00',
       '9/1/11 0:00', '2/1/12 0:00', '1/1/12 0:00', '12/1/12 0:00',
       '9/1/10 0:00', '4/1/12 0:00', '11/1/12 0:00', '5/1/15 0:00',
       '6/1/10 0:00', '2/1/11 0:00', '8/1/10 0:00', '5/1/10 0:00',
       '5/1/11 0:00', '8/1/04 0:00', '11/1/08 0:00', '6/1/09 0:00',
       '2/1/10 0:00', '11/1/09 0:00', '3/1/10 0:00', '11/1/15 0:00',
       '1/1/06 0:00', '8/1/15 0:00', '1/1/10 0:00', '12/1/15 0:00',
       '9/1/07 0:00', '6/1/91 0:00', '7/1/07 0:00', '6/1/07 0:00',
       '2/1/07 0:00'], dtype=object)
```

[14]:
```
df['DOL'].unique()
```

[14]:
```
array(['present', '3/1/15 0:00', '5/1/15 0:00', '7/1/15 0:00',
       '4/1/15 0:00', '10/1/14 0:00', '9/1/14 0:00', '6/1/14 0:00',
       '9/1/12 0:00', '12/1/13 0:00', '6/1/15 0:00', '10/1/13 0:00',
```

```
       '1/1/15 0:00', '4/1/14 0:00', '6/1/13 0:00', '3/1/12 0:00',
       '7/1/14 0:00', '2/1/13 0:00', '1/1/14 0:00', '4/1/13 0:00',
       '7/1/12 0:00', '5/1/14 0:00', '9/1/13 0:00', '2/1/15 0:00',
       '1/1/12 0:00', '8/1/15 0:00', '8/1/14 0:00', '12/1/15 0:00',
       '12/1/14 0:00', '5/1/12 0:00', '3/1/11 0:00', '7/1/11 0:00',
       '2/1/14 0:00', '12/1/11 0:00', '10/1/15 0:00', '11/1/14 0:00',
       '3/1/14 0:00', '11/1/11 0:00', '5/1/13 0:00', '7/1/13 0:00',
       '11/1/13 0:00', '1/1/11 0:00', '5/1/11 0:00', '2/1/12 0:00',
       '11/1/12 0:00', '6/1/12 0:00', '8/1/13 0:00', '3/1/05 0:00',
       '3/1/13 0:00', '10/1/12 0:00', '2/1/11 0:00', '2/1/10 0:00',
       '1/1/13 0:00', '6/1/11 0:00', '9/1/15 0:00', '4/1/12 0:00',
       '8/1/12 0:00', '4/1/11 0:00', '10/1/11 0:00', '11/1/15 0:00',
       '12/1/12 0:00', '9/1/11 0:00', '8/1/10 0:00', '8/1/11 0:00',
       '6/1/09 0:00', '3/1/08 0:00', '10/1/10 0:00'], dtype=object)
```

[15]:
```
df['Designation'].unique()
```

[15]:
```
array(['senior quality engineer', 'assistant manager', 'systems engineer',
       'senior software engineer', 'get', 'system engineer',
       'java software engineer', 'mechanical engineer',
       'electrical engineer', 'project engineer', 'senior php developer',
       'senior systems engineer', 'quality assurance engineer',
       'qa analyst', 'network engineer', 'product development engineer',
       'associate software developer', 'data entry operator',
       'software engineer', 'developer', 'electrical project engineer',
       'programmer analyst', 'systems analyst', 'ase',
       'telecommunication engineer', 'application developer',
       'ios developer', 'executive assistant', 'online marketing manager',
       'documentation specialist', 'associate software engineer',
       'management trainee', 'site manager', 'software developer',
       '.net developer', 'production engineer', 'jr. software engineer',
       'trainee software developer', 'ui developer',
       'assistant system engineer', 'android developer',
       'customer service', 'test engineer', 'java developer', 'engineer',
       'recruitment coordinator', 'technical support engineer',
       'data analyst', 'assistant software engineer', 'faculty',
       'entry level management trainee',
       'customer service representative', 'software test engineer',
       'firmware engineer', 'php developer', 'research associate',
       'research analyst', 'quality engineer', 'programmer',
       'technical support executive', 'business analyst', 'web developer',
       'application engineer', 'project coordinator', 'engineer trainee',
       'sap consultant', 'quality analyst', 'marketing coordinator',
       'system administrator', 'senior engineer',
       'business development managerde', 'network administrator',
       'technical support specialist', 'business development executive',
       'junior software engineer', 'asp.net developer',
```

```
'graduate engineer trainee', 'field engineer',
'assistant professor', 'trainee software engineer',
'senior software developer',
'quality assurance automation engineer', 'design engineer',
'telecom engineer', 'quality control engineer',
'hardware engineer', 'hr recruiter', 'sales associate',
'junior engineer', 'associate engineer', 'maintenance engineer',
'sales engineer', 'human resources associate',
'mobile application developer',
'electronic field service engineer', 'process associate',
'field service engineer', 'it support specialist',
'software development engineer', 'business process analyst',
'operation engineer', 'electrical designer', 'marketing assistant',
'sales executive', 'admin assistant', 'senior java developer',
'account executive', 'oracle dba', 'rf engineer',
'embedded software engineer', 'programmer analyst trainee',
'technical engineer', 'operations executive', 'trainee engineer',
'recruiter', 'lecturer', '.net web developer',
'marketing executive', 'operations assistant', 'associate manager',
'electrical design engineer', 'systems administrator',
'client services associate', 'it analyst', 'senior developer',
'cad designer', 'business technology analyst', 'asst. manager',
'service engineer', 'executive recruiter', 'planning engineer',
'associate technical operations', 'web designer',
'software architect', 'software quality assurance tester',
'seo trainee', 'process engineer',
'software quality assurance analyst', 'designer',
'business systems consultant', 'business development manager',
'junior research fellow', 'technical recruiter',
'operations analyst', 'quality assurance test engineer',
'linux systems administrator', 'software trainee',
'entry level sales and marketing', 'electrical field engineer',
'windows systems administrator', 'junior software developer',
'python developer', 'web application developer',
'assistant systems engineer', 'javascript developer',
'operation executive', 'performance engineer', 'technical writer',
'operations engineer and jetty handling', 'lead engineer',
'portfolio analyst', 'associate system engineer',
'mechanical design engineer', 'product engineer',
'network security engineer', 'operations manager',
'technical lead', 'operations', 'quality assurance tester',
'automation engineer', 'data scientist', 'quality associate',
'manual tester', 'sr. engineer', 'embedded engineer',
'service and sales engineer', 'telecom support engineer',
'engineer- customer support', 'cloud engineer', 'branch manager',
'business analyst consultant', 'technology lead',
'software trainee engineer', 'dcs engineer', 'junior manager',
```

```
'ux designer', 'clerical', 'hr generalist',
'database administrator', 'senior design engineer', 'seo',
'assistant engineer', 'marketing analyst', 'it executive',
'salesforce developer', 'software tester', 'sql dba',
'junior engineer product support', 'manager',
'senior business analyst', 'c# developer',
'implementation engineer', 'executive hr', 'executive engineer',
'sharepoint developer', 'system analyst',
'sales management trainee', 'senior project engineer',
'it recruiter', 'software engineer analyst',
'desktop support technician', 'continuous improvement engineer',
'process advisor', 'etl developer', 'sales and service engineer',
'project manager', 'training specialist', 'product manager',
'staffing recruiter', 'assistant programmer', 'quality controller',
'mis executive', 'game developer', 'digital marketing specialist',
'principal software engineer', 'software devloper',
'senior mechanical engineer', 'technical operations analyst',
'service coordinator', 'testing engineer', 'technical assistant',
'sap abap consultant', 'seo engineer', 'project assistant',
'talent acquisition specialist', 'sales account manager',
'software engineer trainee', 'customer service manager',
'help desk analyst', 'general manager', 'engineering manager',
'senior network engineer',
'field based employee relations manager', 'phone banking officer',
'support engineer', 'associate test engineer',
'technology analyst', 'network support engineer',
'it business analyst', 'junior system analyst',
'senior .net developer', 'secretary', 'research engineer',
'quality assurance auditor', 'process executive',
'lecturer & electrical maintenance', 'office coordinator',
'hr manager', 'html developer', 'sales support',
'front end web developer', 'administrative support',
'territory sales manager', 'project administrator',
'environmental engineer', 'web designer and seo',
'information security analyst',
'field business development associate', 'operational executive',
'administrative coordinator', 'senior risk consultant',
'desktop support engineer', 'cad drafter', 'noc engineer',
'industrial engineer', 'it engineer', 'human resources intern',
'senior quality assurance engineer', 'clerical assistant',
'software enginner', 'quality assurance',
'delivery software engineer', 'graphic designer',
'sales development manager', 'visiting faculty',
'business intelligence analyst', 'team lead',
'operational excellence manager', 'sales & service engineer',
'web intern', 'full stack developer', 'database developer',
'sr. database engineer', 'graduate apprentice trainee',
```

```
'software engineer associate', 'technical analyst',
'executive engg', 'it technician', 'business system analyst',
'process control engineer', 'technical consultant',
'business office manager', 'quality control inspector',
'product design engineer', 'manufacturing engineer',
'seo executive', 'sap analyst', 'software engineere',
'financial service consultant', 'co faculty', 'software analyst',
'desktop support analyst', 'graduate engineer',
'engineering technician', 'it assistant', 'marketing manager',
'human resource assistant', 'hr assistant', 'product developer',
'customer support engineer',
'quality control inspection technician', 'gis/cad engineer',
'senior web developer', 'sql developer', 'research staff member',
'sap abap associate consultant', 'associate qa',
'corporate recruiter', 'project management officer',
'business systems analyst', 'software programmer',
'help desk technician', 'sales manager', 'catalog associate',
'assistant store manager', 'software engg', 'it developer',
'apprentice', 'business consultant', 'controls engineer',
'ruby on rails developer', 'risk consultant', 'account manager',
'professor', 'assistant administrator', 'civil engineer',
'educator', 'service manager', 'teradata dba',
'full-time loss prevention associate', 'junior recruiter',
'associate developer', 'assistant electrical engineer',
'shift engineer', 'dotnet developer', 'rf/dt engineer',
'human resources analyst', 'software test engineerte',
'junior .net developer', 'java trainee', 'maintenance supervisor',
'r&d engineer', 'front end developer', 'engineer-hws',
'operations engineer', 'senior research fellow',
'web designer and joomla administrator',
'enterprise solutions developer',
'information technology specialist', 'site engineer',
'graduate trainee engineer', 'quality assurance analyst',
'cnc programmer', 'financial analyst', 'system engineer trainee',
'sap mm consultant', 'assistant system engineer trainee',
'qa trainee', 'teradata developer', 'hr executive',
'senior programmer', 'software test engineer (etl)',
'associate software engg', 'supply chain analyst', 'sales trainer',
'software executive', 'team leader',
'assistant system engineer - trainee', 'seo analyst',
'risk investigator', 'executive administrative assistant',
'program manager', 'r & d', 'sap functional consultant',
'website developer/tester', 'software designer',
'sales coordinator', 'qa engineer', 'aircraft technician',
'customer care executive', 'senior test engineer',
'program analyst trainee', 'electrical controls engineer',
'trainee decision scientist', 'editor', 'bss engineer', 'dba',
```

```
       'software eng', 'computer faculty', 'recruitment associate',
       'logistics executive', 'quality consultant',
       'senior sales executive', 'db2 dba', 'test technician',
       'it operations associate', 'software engineering associate',
       'research scientist', 'jr. software developer'], dtype=object)
```

[16]:
```
df['JobCity'].unique()
```

[16]:
```
array(['Bangalore', 'Indore', 'Chennai', 'Gurgaon', 'Manesar',
       'Hyderabad', 'Banglore', 'Noida', 'Kolkata', 'Pune', '-1',
       'mohali', 'Jhansi', 'Delhi', 'Hyderabad ', 'Bangalore ', 'noida',
       'delhi', 'Bhubaneswar', 'Navi Mumbai', 'Mumbai', 'New Delhi',
       'Mangalore', 'Rewari', 'Gaziabaad', 'Bhiwadi', 'Mysore', 'Rajkot',
       'Greater Noida', 'Jaipur', 'noida ', 'HYDERABAD', 'mysore',
       'THANE', 'Maharajganj', 'Thiruvananthapuram', 'Punchkula',
       'Bhubaneshwar', 'Pune ', 'coimbatore', 'Dhanbad', 'Lucknow',
       'Trivandrum', 'kolkata', 'mumbai', 'Gandhi Nagar', 'Una',
       'Daman and Diu', 'chennai', 'GURGOAN', 'vsakhapttnam', 'pune',
       'Nagpur', 'Bhagalpur', 'new delhi - jaisalmer', 'Coimbatore',
       'Ahmedabad', 'Kochi/Cochin', 'Bankura', 'Bengaluru', 'Mysore ',
       'Kanpur ', 'jaipur', 'Gurgaon ', 'bangalore', 'CHENNAI',
       'Vijayawada', 'Kochi', 'Beawar', 'Alwar', 'NOIDA', 'Greater noida',
       'Siliguri ', 'raipur', 'gurgaon', 'Bhopal', 'Faridabad', 'Jodhpur',
       'udaipur', 'Muzaffarpur', 'Kolkata`', 'Bulandshahar', 'Haridwar',
       'Raigarh', 'Visakhapatnam', 'Jabalpur', 'hyderabad', 'Unnao',
       'KOLKATA', 'Thane', 'Aurangabad', 'Belgaum', 'gurgoan', 'Dehradun',
       'Rudrapur', 'Jamshedpur', 'vizag', 'Nouda', 'Dharamshala',
       'Banagalore', 'Hissar', 'Ranchi', 'BANGALORE', 'Madurai', 'Gurga',
       'Chandigarh', 'Australia', ' Chennai', 'CHEYYAR', 'Mumbai ',
       'sonepat', 'Ghaziabad', 'Pantnagar', 'Siliguri', 'mumbai ',
       'Jagdalpur', 'Chennai ', 'angul', 'Baroda', ' ariyalur', 'Jowai',
       'Kochi/Cochin, Chennai and Coimbatore', 'bhubaneswar', 'Neemrana',
       'VIZAG', 'Tirupathi', 'Lucknow ', 'Ahmedabad ', 'Bhubneshwar',
       'Noida ', 'pune ', 'Calicut', 'Gandhinagar', 'LUCKNOW', 'Dubai',
       'bengaluru', 'MUMBAI', 'Ahmednagar', 'Nashik', 'New delhi',
       'Bellary', 'Ludhiana', 'New Delhi ', 'Muzaffarnagar', 'BHOPAL',
       'Gurgoan', 'Gagret', 'Indirapuram, Ghaziabad', 'Gwalior',
       'new delhi', 'TRIVANDRUM', 'Chennai & Mumbai', 'Rajasthan',
       'Sonipat', 'Bareli', 'Kanpur', 'Hospete', 'Miryalaguda', ' mumbai',
       'Dharuhera', 'lucknow', 'meerut', 'dehradun', 'Ganjam', 'Hubli',
       'bangalore ', 'NAVI MUMBAI', 'ncr', 'Agra', 'Trichy',
       'kudankulam ,tarapur', 'Ongole', 'Sambalpur', 'Pondicherry',
       'Bundi', 'SADULPUR,RAJGARH,DISTT-CHURU,RAJASTHAN', 'AM', 'Bikaner',
       'Vadodara', 'BAngalore', 'india', 'Asansol', 'Tirunelvelli',
       'Ernakulam', 'DELHI', 'Bilaspur', 'Chandrapur', 'Nanded',
       'Dharmapuri', 'Vandavasi', 'Rohtak', 'trivandrum', 'Nagpur ',
       'Udaipur', 'Patna', 'banglore', 'indore', 'Salem', 'Nasikcity',
```

```
       'Gandhinagar ', 'Technopark, Trivandrum', 'Bharuch', 'Tornagallu',
       'Raipur', 'Kolkata ', 'Jaspur', 'Burdwan', 'Bhubaneswar ',
       'Shimla', 'ahmedabad', 'Gajiabaad', 'Jammu', 'Shahdol',
       'Muvattupuzha', 'Al Jubail,Saudi Arabia', 'Kalmar, Sweden',
       'Secunderabad', 'A-64,sec-64,noida', 'Ratnagiri', 'Jhajjar',
       'Gulbarga', 'hyderabad(bhadurpally)', 'Nalagarh', 'Chandigarh ',
       'Jaipur ', 'Jeddah Saudi Arabia', ' Delhi', 'PATNA', 'SHAHDOL',
       'Chennai, Bangalore', 'Bhopal ', 'Jamnagar', 'PUNE', 'Tirupati',
       'Gonda', 'jamnagar', 'chennai ', 'orissa', 'kharagpur',
       'Trivandrum ', 'Navi Mumbai , Hyderabad', 'Joshimath',
       'chandigarh', 'Bathinda', 'Johannesburg', 'kala amb ', 'Karnal',
       'LONDON', 'Kota', 'Panchkula', 'Baddi HP', 'Nagari',
       'Mettur, Tamil Nadu ', 'Durgapur', 'pondi', 'Surat', 'Kurnool',
       'kolhapur', 'Madurai ', 'GREATER NOIDA', 'Bhilai', ' Pune',
       'hderabad', 'KOTA', 'thane', 'Vizag', 'Bahadurgarh',
       'Rayagada, Odisha', 'kakinada', 'GURGAON', 'Varanasi', 'punr',
       'Nellore', 'patna', 'Meerut', 'hyderabad ', 'Sahibabad', 'Howrah',
       'BHUBANESWAR', 'Trichur', 'Ambala', 'Khopoli', 'keral', 'Roorkee',
       'Greater NOIDA', 'Navi mumbai', 'ghaziabad', 'Allahabad',
       'Delhi/NCR', 'Panchkula ', 'Ranchi ', 'Jalandhar', 'manesar',
       'vapi', 'PILANI', 'muzzafarpur', 'RAS AL KHAIMAH', 'bihar',
       'singaruli', 'KANPUR', 'Banglore ', 'pondy', 'Mohali', 'Phagwara',
       ' Mumbai', ' bangalore', 'GURAGAON', 'Baripada', 'MEERUT',
       'Yamuna Nagar', 'shahibabad', 'sampla', 'Guwahati', 'Rourkela',
       'Banaglore', 'Vellore', 'Dausa', 'latur (Maharashtra )',
       'NEW DELHI', 'kanpur', 'Mainpuri', 'karnal', 'Dammam', 'Haldia',
       'sambalpur', 'RAE BARELI', 'ranchi', 'jAipur', 'BANGLORE',
       'Patiala', 'Gorakhpur', 'new dehli', 'BANGALORE ', 'Ambala City',
       'Karad', 'Rajpura', 'Pilani', 'haryana', 'Asifabadbanglore'],
      dtype=object)
```

[17]:
```
df['Gender'].unique()
```

[17]:
```
array(['f', 'm'], dtype=object)
```

[18]:
```
df['DOB'].unique()
```

[18]:
```
array(['2/19/90 0:00', '10/4/89 0:00', '8/3/92 0:00', …, '2/28/86 0:00',
       '6/22/90 0:00', '4/15/87 0:00'], dtype=object)
```

[19]:
```
df['10percentage'].unique()
```

[19]:
```
array([84.3 , 85.4 , 85.  , 85.6 , 78.  , 89.92, 86.08, 92.  , 90.  ,
       77.  , 88.6 , 81.  , 83.6 , 60.8 , 81.2 , 69.54, 85.8 , 65.  ,
       79.  , 91.2 , 75.67, 92.5 , 70.  , 88.  , 86.8 , 90.88, 76.8 ,
       84.  , 64.  , 77.2 , 87.  , 73.  , 71.  , 91.5 , 52.33, 66.6 ,
       75.  , 91.4 , 59.  , 84.5 , 95.8 , 69.6 , 86.6 , 81.92, 66.5 ,
```

80. , 85.2 , 89.25, 58.4 , 90.8 , 89.88, 63.3 , 72. , 83. ,
85.3 , 60.66, 89.37, 84.4 , 89. , 86. , 70.6 , 93.44, 76. ,
86.4 , 84.83, 72.3 , 69. , 73.33, 86.16, 56. , 79.25, 88.66,
80.8 , 81.16, 83.5 , 55.6 , 84.6 , 62. , 68.4 , 51. , 79.6 ,
63.4 , 86.53, 76.18, 91.43, 76.17, 82.4 , 58. , 90.04, 60.4 ,
74.23, 59.8 , 82.3 , 84.2 , 60. , 82.8 , 82.6 , 74. , 89.8 ,
88.67, 64.66, 77.8 , 89.5 , 64.56, 91.12, 73.08, 78.33, 81.33,
89.7 , 87.4 , 92.66, 76.87, 91.3 , 77.9 , 76.2 , 77.78, 65.6 ,
65.8 , 67.75, 91.6 , 78.66, 78.4 , 61. , 90.4 , 58.6 , 82.2 ,
82. , 90.1 , 86.17, 83.89, 76.7 , 88.2 , 80.6 , 91. , 74.4 ,
79.28, 84.96, 92.8 , 79.4 , 66.8 , 79.8 , 65.3 , 94.6 , 83.33,
80.83, 67.23, 86.2 , 55. , 86.62, 69.13, 89.12, 87.5 , 66.4 ,
80.33, 75.2 , 50.6 , 81.1 , 60.14, 78.83, 75.8 , 77.66, 64.48,
67. , 75.6 , 85.28, 71.5 , 93.6 , 93.33, 78.8 , 82.16, 77.65,
56.5 , 79.83, 57. , 89.42, 72.8 , 86.3 , 77.5 , 71.2 , 80.2 ,
73.6 , 68. , 74.7 , 69.2 , 65.33, 96.48, 82.5 , 91.8 , 93.4 ,
68.5 , 73.4 , 72.2 , 71.8 , 66.33, 71.67, 70.2 , 90.27, 67.67,
53.4 , 76.16, 65.71, 79.5 , 87.2 , 80.4 , 88.4 , 90.6 , 89.2 ,
81.6 , 86.5 , 77.16, 72.33, 79.89, 75.4 , 72.83, 88.33, 78.88,
95.2 , 89.33, 75.3 , 69.8 , 52. , 58.2 , 90.5 , 70.83, 62.13,
74.5 , 63. , 73.37, 83.4 , 92.6 , 78.6 , 76.4 , 86.88, 66. ,
70.67, 65.1 , 68.14, 92.2 , 93.5 , 82.83, 81.66, 90.15, 44.16,
94.4 , 64.4 , 85.17, 70.1 , 88.25, 94.8 , 77.4 , 66.66, 81.03,
44. , 45.6 , 87.8 , 72.6 , 79.86, 84.67, 48. , 53.3 , 71.66,
68.8 , 78.15, 83.43, 86.9 , 84.8 , 75.06, 86.66, 70.9 , 81.12,
67.5 , 78.2 , 71.06, 71.28, 62.1 , 90.56, 87.6 , 71.3 , 88.8 ,
86.09, 67.72, 76.6 , 68.6 , 63.43, 70.4 , 67.6 , 73.8 , 55.5 ,
74.67, 92.16, 83.66, 62.86, 49. , 87.11, 77.53, 88.5 , 61.9 ,
79.2 , 83.8 , 79.33, 67.3 , 83.2 , 72.5 , 77.67, 94.2 , 59.33,
87.63, 89.76, 84.14, 88.17, 59.6 , 64.3 , 75.04, 82.9 , 73.5 ,
73.67, 77.7 , 87.69, 83.16, 71.32, 61.6 , 61.8 , 85.06, 91.71,
75.46, 74.8 , 67.4 , 90.3 , 86.15, 64.7 , 69.7 , 82.33, 58.83,
75.36, 76.5 , 66.67, 67.2 , 80.88, 88.88, 71.4 , 65.2 , 96. ,
61.2 , 81.4 , 68.66, 65.56, 86.7 , 63.2 , 62.2 , 68.2 , 77.85,
83.3 , 56.78, 83.04, 71.1 , 76.53, 74.83, 59.3 , 73.2 , 86.33,
64.83, 72.1 , 61.1 , 86.83, 70.8 , 94. , 80.3 , 64.6 , 84.1 ,
84.93, 92.83, 89.6 , 80.5 , 83.82, 77.57, 51.2 , 66.46, 82.67,
61.4 , 69.4 , 90.24, 75.75, 90.83, 89.1 , 87.33, 83.1 , 88.34,
91.67, 87.3 , 48.8 , 75.38, 55.52, 71.05, 77.63, 92.47, 93. ,
68.33, 79.16, 85.33, 80.42, 78.25, 95.52, 87.86, 89.4 , 94.7 ,
97.12, 93.94, 59.57, 80.53, 81.9 , 63.6 , 66.16, 62.5 , 69.5 ,
80.93, 82.7 , 79.29, 81.5 , 62.34, 83.68, 70.66, 60.1 , 77.38,
68.83, 94.43, 85.53, 88.09, 74.2 , 85.73, 72.4 , 67.7 , 79.78,
81.3 , 79.37, 85.5 , 87.83, 70.33, 80.64, 58.7 , 60.2 , 77.81,
85.67, 89.9 , 79.75, 75.73, 87.68, 60.5 , 81.38, 75.52, 48.5 ,
88.3 , 82.1 , 85.18, 81.8 , 90.46, 70.5 , 79.52, 89.29, 61.75,
78.67, 71.6 , 80.13, 81.67, 95. , 81.14, 72.16, 87.04, 88.64,

```
82.56, 90.01, 63.66, 65.17, 91.84, 92.1 , 43.  , 65.23, 92.48,
82.88, 73.07, 58.56, 85.83, 67.34, 85.75, 80.7 , 79.23, 70.26,
52.7 , 75.86, 90.16, 90.2 , 78.5 , 58.9 , 80.32, 67.16, 73.06,
92.93, 85.76, 87.52, 88.36, 81.73, 60.7 , 87.7 , 79.85, 68.9 ,
73.83, 64.8 , 77.84, 74.14, 64.28, 92.4 , 73.94, 63.33, 70.06,
88.7 , 89.3 , 83.83, 91.33, 80.07, 72.17, 63.5 , 69.17, 67.42,
84.16, 76.64, 78.13, 61.69, 81.7 , 82.46, 64.57, 78.1 , 66.3 ,
59.71, 80.67, 77.88, 88.53, 93.38, 52.93, 78.17, 72.36, 84.75,
86.67, 77.6 , 74.3 , 62.4 , 65.16, 59.2 , 69.85, 79.68, 65.4 ,
94.72, 71.86, 81.25, 87.62, 54.  , 85.92, 74.33, 82.28, 75.1 ,
69.73, 92.12, 70.3 , 76.33, 80.1 , 76.67, 77.83, 57.67, 83.14,
50.  , 91.21, 81.83, 78.16, 80.14, 93.3 , 61.63, 73.73, 80.46,
76.48, 82.25, 56.16, 57.88, 87.07, 67.1 , 87.73, 77.12, 64.53,
86.46, 82.62, 53.06, 74.62, 76.66, 45.33, 69.69, 53.  , 75.33,
74.28, 74.6 , 74.88, 74.53, 46.24, 80.15, 86.13, 85.72, 51.36,
78.53, 53.8 , 86.1 , 84.43, 76.36, 77.86, 88.83, 88.18, 79.14,
54.5 , 84.33, 78.3 , 77.44, 87.31, 58.16, 61.3 , 77.33, 75.12,
72.12, 65.26, 83.56, 50.5 , 82.27, 67.36, 87.16, 75.83, 78.44,
91.9 , 68.3 , 92.64, 58.17, 83.46, 88.04, 70.58, 71.17, 84.13,
64.62, 66.15, 67.8 , 57.78, 56.2 , 61.5 , 94.1 , 74.16, 78.93,
70.56, 85.16, 82.26, 71.13, 65.66, 71.71, 86.37, 88.57, 59.05,
79.66, 89.39, 95.54, 55.3 , 70.14, 87.23, 85.38, 86.92, 95.04,
95.6 , 60.83, 71.33, 94.16, 78.57, 80.16, 70.25, 82.13, 88.76,
51.6 , 70.76, 57.5 , 68.67, 74.18, 68.25, 71.04, 64.43, 82.24,
90.06, 67.12, 75.85, 87.81, 76.93, 65.5 , 92.3 , 50.66, 61.83,
63.16, 64.2 , 53.6 , 83.69, 80.04, 78.86, 70.61, 62.8 , 67.06,
65.85, 94.24, 63.8 , 75.77, 94.5 , 71.83, 91.1 , 91.52, 90.08,
93.16, 86.34, 88.1 , 97.  , 62.93, 76.96, 85.46, 78.08, 66.7 ,
95.86, 92.09, 81.46, 81.86, 61.73, 77.22, 46.8 , 79.3 , 72.75,
93.8 , 93.67, 75.16, 72.45, 85.62, 86.85, 93.28, 58.33, 64.33,
75.62, 54.8 , 58.5 , 91.68, 69.3 , 62.6 , 65.41, 91.62, 88.16,
92.32, 69.83, 83.67, 69.92, 52.5 , 56.4 , 54.6 , 78.61, 69.53,
75.71, 71.84, 70.16, 69.66, 95.4 , 84.62, 91.53, 85.66, 61.57,
80.63, 69.33, 62.3 , 87.87, 70.75, 87.58, 58.8 , 62.88, 73.66,
97.76, 88.44, 54.83, 67.25, 90.76, 82.75, 75.66, 91.04, 90.58,
91.86, 73.1 , 73.3 , 69.1 , 51.83, 62.26, 65.67, 68.65, 51.42,
78.28, 80.58, 55.33, 91.17, 74.9 , 93.43, 90.81, 89.89, 62.67,
90.26, 62.15, 70.08, 87.88, 72.67, 93.2 , 60.46, 71.07, 46.  ,
54.4 , 86.12, 72.15, 71.85, 49.9 , 83.75, 71.73, 90.33, 86.14,
66.2 , 88.75, 67.33, 57.14, 67.76, 82.66, 77.92, 79.38, 67.17,
89.17, 79.67, 96.8 , 71.37, 82.87, 89.44, 71.76, 57.7 , 89.23,
79.06, 83.25, 61.86, 89.56, 82.14, 70.27, 59.7 , 93.07, 79.9 ,
64.5 , 66.85, 69.16, 52.09, 78.72])
```

[20]:
```
df['10board'].unique()
```

[20]:
```
array(['board ofsecondary education,ap', 'cbse', 'state board',
       'mp board bhopal', 'icse',
       'karnataka secondary school of examination', 'up',
       'karnataka state education examination board', 'ssc',
       'kerala state technical education', '0', 'bseb',
       'state board of secondary education, andhra pradesh',
       'matriculation', 'gujarat state board', 'karnataka state board',
       'wbbse', 'maharashtra state board', 'icse board', 'up board',
       'board of secondary education(bse) orissa',
       'little jacky matric higher secondary school',
       'uttar pradesh board', 'bsc,orissa', 'mp board', 'upboard',
       'matriculation board', 'j & k bord', 'rbse',
       'central board of secondary education', 'pseb', 'jkbose',
       'haryana board of school education,(hbse)', 'metric', 'ms board',
       'kseeb', 'stateboard', 'maticulation',
       'karnataka secondory education board', 'mumbai board', 'sslc',
       'kseb', 'board secondary  education', 'matric board',
       'board of secondary education',
       'west bengal board of secondary education',
       'jharkhand secondary examination board,ranchi', 'u p board',
       'bseb,patna', 'hsc', 'bse', 'sss pune',
       'karnataka education board (keeb)', 'kerala',
       'state board of secondary education( ssc)', 'gsheb',
       'up(allahabad)', 'nagpur', 'don bosco maatriculation school',
       'karnataka state secondary education board', 'maharashtra',
       'karnataka secondary education board',
       'himachal pradesh board of school education',
       'certificate of middle years program of ib',
       'karnataka board of secondary education',
       'board of secondary education rajasthan', 'uttarakhand board',
       'ua', 'board of secendary education orissa',
       'karantaka secondary education and examination borad', 'hbsc',
       'kseeb(karnataka secondary education examination board)',
       'cbse[gulf zone]', 'hbse', 'state(karnataka board)',
       'jharkhand accademic council',
       'jharkhand secondary examination board (ranchi)',
       'karnataka secondary education examination board', 'delhi board',
       'mirza ahmed ali baig', 'jseb', 'bse, odisha', 'bihar board',
       'maharashtra state(latur board)', 'rajasthan board', 'mpboard',
       'upbhsie', 'secondary board of rajasthan',
       'tamilnadu matriculation board', 'jharkhand secondary board',
       'board of secondary education,andhara pradesh', 'up baord',
       'state', 'board of intermediate education',
       'state board of secondary education,andhra pradesh',
       'up board , allahabad',
       'stjosephs girls higher sec school,dindigul', 'maharashtra board',
       'education board of kerala', 'board of ssc',
```

```
        'maharashtra state board pune',
        'board of school education harayana',
        'secondary school cerficate', 'maharashtra sate board', 'ksseb',
        'bihar examination board, patna', 'latur',
        'board of secondary education, rajasthan', 'state borad hp',
        'cluny', 'bsepatna', 'up borad', 'ssc board of andrapradesh',
        'matric', 'bse,orissa', 'ssc-andhra pradesh', 'mp',
        'karnataka education board', 'mhsbse',
        'karnataka sslc board bangalore', 'karnataka', 'u p',
        'secondary school of education', 'state board of karnataka',
        'karnataka secondary board', 'andhra pradesh board ssc',
        'stjoseph of cluny matrhrsecschool,neyveli,cuddalore district',
        'hse,orissa', 'national public school', 'nagpur board',
        'jharkhand academic council', 'bsemp',
        'board of secondary education, andhra pradesh',
        'board of secondary education orissa',
        'board of secondary education,rajasthan(rbse)',
        'board of secondary education,ap',
        'board of secondary education,andhra pradesh',
        'jawahar navodaya vidyalaya', 'aisse',
        'karnataka board of higher education', 'bihar',
        'kerala state board', 'cicse', 'tn state board',
        'kolhapur divisional board, maharashtra',
        'bharathi matriculation school', 'uttaranchal state board',
        'wbbsce', 'mp state board', 'seba(assam)', 'anglo indian', 'gseb',
        'uttar pradesh', 'ghseb', 'board of school education uttarakhand',
        'msbshse,pune', 'tamilnadu state board', 'kerala university',
        'uttaranchal shiksha avam pariksha parishad',
        'bse(board of secondary education)',
        'bright way college, (up board)',
        'school secondary education, andhra pradesh',
        'secondary state certificate',
        'maharashtra state board of secondary and higher secondary
education,pune',
        'andhra pradesh state board', 'stmary higher secondary', 'cgbse',
        'secondary school certificate', 'rajasthan board ajmer', 'mpbse',
        'pune board', 'cbse ', 'board of secondary education,orissa',
        'maharashtra state board,pune', 'up bord',
        'kiran english medium high school', 'state board (jac, ranchi)',
        'gujarat board', 'state board ', 'sarada high scchool',
        'kalaimagal matriculation higher secondary school',
        'karnataka board', 'maharastra board', 'sslc board',
        'ssc maharashtra board', 'tamil nadu state', 'uttrakhand board',
        'bihar secondary education board,patna',
        'haryana board of school education',
        'sri kannika parameswari highier secondary school, udumalpet',
        'ksseb(karnataka state board)', 'nashik board',
```

```
 'jharkhand secondary education board', 'himachal pradesh board',
 'maharashtra satate board',
 'maharashtra state board mumbai divisional board',
 'dav public school,hehal',
 'state board of secondary education, ap',
 'rajasthan board of secondary education', 'hsce',
 'karnataka secondary education',
 'board of secondary education,odisha', 'maharashtra nasik board',
 'west bengal board of secondary examination (wbbse)',
 'holy cross matriculation hr sec school', 'cbsc', 'apssc',
 'bseb patna', 'kolhapur', 'bseb, patna', 'up board allahabad',
 'biharboard', 'nagpur board,nagpur', 'pune', 'gyan bharati school',
 'rbse,ajmer', 'board of secondaray education',
 'secondary school education', 'state bord', 'jbse,jharkhand',
 'hse', 'madhya pradesh board', 'bihar school examination board',
 'west bengal board of secondary eucation', 'state boardmp board ',
 'icse board , new delhi',
 'board of secondary education (bse) orissa',
 'maharashtra state board for ssc',
 'board of secondary school education', 'latur board',
 "stmary's convent inter college", 'nagpur divisional board',
 'ap state board', 'cgbse raipur', 'uttranchal board', 'ksbe',
 'central board of secondary education, new delhi',
 'bihar school examination board patna', 'cbse board',
 'sslc,karnataka', 'mp-bse', 'up bourd', 'dav public school sec 14',
 'board of school education haryana',
 'council for indian school certificate examination',
 'aurangabad board', 'j&k state board of school education',
 'maharashtra state board of secondary and higher secondary education',
 'maharashtra state boar of secondary and higher secondary education',
 'ssc regular', 'karnataka state examination board', 'nasik',
 'west bengal  board of secondary education', 'up board,allahabad',
 'bseb ,patna',
 'state board - west bengal board of secondary education : wbbse',
 'maharashtra state board of secondary & higher secondary education',
 'delhi public school', 'karnataka secondary eduction',
 'secondary education board of rajasthan',
 'maharashtra board, pune', 'rbse (state board)', 'apsche',
 'board of  secondary education',
 'board of high school and intermediate education uttarpradesh',
 'kea', 'board of secondary education - andhra pradesh',
 'ap state board for secondary education', 'seba',
 'punjab school education board, mohali',
 'jharkhand acedemic council', 'hse,board',
 'board of ssc education andhra pradesh', 'up-board', 'bse,odisha'],
dtype=object)
```

[21]:
```
df['12graduation'].unique()
```

[21]:
```
array([2007, 2010, 2008, 2009, 2006, 2011, 2005, 1995, 2004, 2012, 2003,
       2002, 2001, 1998, 2013, 1999])
```

[22]:
```
df['12percentage'].unique()
```

[22]:
```
array([95.8 , 85.  , 68.2 , 83.6 , 76.8 , 87.  , 67.5 , 91.  , 91.2 ,
       72.2 , 83.7 , 86.  , 69.83, 62.4 , 79.9 , 64.43, 74.8 , 66.66,
       64.8 , 62.2 , 84.63, 74.4 , 95.2 , 43.42, 90.  , 82.8 , 82.5 ,
       83.  , 68.  , 74.  , 92.  , 86.1 , 84.4 , 68.4 , 61.  , 93.8 ,
       85.4 , 67.  , 89.66, 68.6 , 60.  , 73.  , 87.7 , 87.16, 82.  ,
       71.4 , 75.  , 61.46, 49.5 , 72.5 , 81.5 , 78.  , 90.1 , 70.1 ,
       95.4 , 61.1 , 49.  , 79.  , 88.3 , 77.8 , 94.4 , 86.67, 73.2 ,
       64.  , 77.  , 78.67, 72.  , 78.2 , 95.  , 82.4 , 60.2 , 62.6 ,
       70.  , 71.33, 70.08, 56.  , 80.  , 84.33, 87.9 , 65.  , 68.5 ,
       94.2 , 66.  , 88.  , 76.6 , 50.8 , 66.5 , 76.4 , 65.8 , 48.  ,
       63.  , 71.55, 86.33, 71.3 , 57.6 , 83.4 , 75.16, 77.77, 60.25,
       89.6 , 86.4 , 80.4 , 59.  , 73.6 , 63.6 , 66.6 , 86.8 , 79.6 ,
       87.58, 81.4 , 89.  , 62.  , 47.  , 77.2 , 71.2 , 54.  , 67.6 ,
       91.9 , 63.1 , 69.  , 68.46, 85.2 , 78.4 , 82.2 , 95.6 , 81.33,
       88.9 , 82.75, 72.6 , 66.67, 70.2 , 61.5 , 70.6 , 79.4 , 61.8 ,
       95.5 , 80.7 , 60.4 , 77.14, 75.2 , 81.2 , 80.8 , 88.88, 83.9 ,
       65.2 , 83.1 , 80.6 , 70.16, 90.91, 84.7 , 68.55, 59.16, 78.83,
       59.9 , 75.6 , 87.8 , 79.2 , 80.3 , 82.1 , 65.5 , 84.3 , 64.4 ,
       91.6 , 95.3 , 69.8 , 86.9 , 73.4 , 56.9 , 86.7 , 64.7 , 80.5 ,
       61.6 , 90.7 , 81.  , 57.  , 92.6 , 78.6 , 71.  , 71.5 , 70.4 ,
       89.5 , 76.66, 80.1 , 54.4 , 80.9 , 84.8 , 69.45, 93.4 , 56.8 ,
       91.5 , 90.67, 64.5 , 96.1 , 62.8 , 94.1 , 89.7 , 76.  , 73.8 ,
       78.3 , 96.6 , 60.17, 75.4 , 72.4 , 52.  , 77.4 , 69.6 , 56.2 ,
       78.43, 82.7 , 74.45, 76.2 , 68.8 , 78.8 , 50.  , 67.4 , 95.41,
       84.  , 71.08, 94.5 , 67.75, 87.1 , 81.8 , 80.2 , 83.58, 62.83,
       69.4 , 86.5 , 67.67, 74.2 , 66.4 , 55.  , 61.2 , 64.6 , 43.12,
       61.57, 84.5 , 83.8 , 89.1 , 74.6 , 81.1 , 80.83, 67.8 , 76.5 ,
       87.83, 69.9 , 88.4 , 58.55, 83.33, 82.6 , 69.2 , 63.2 , 88.7 ,
       62.66, 61.7 , 96.7 , 62.5 , 79.57, 93.  , 78.16, 74.92, 94.16,
       51.4 , 97.1 , 94.9 , 87.4 , 63.4 , 87.66, 65.4 , 68.67, 58.  ,
       68.66, 67.7 , 91.3 , 79.8 , 81.6 , 67.25, 55.2 , 65.6 , 64.3 ,
       51.63, 93.7 , 69.04, 50.2 , 96.5 , 77.7 , 90.2 , 72.67, 73.83,
       71.9 , 82.83, 82.08, 77.38, 71.8 , 66.8 , 71.25, 53.  , 76.62,
       72.77, 79.83, 77.3 , 81.17, 81.12, 62.16, 67.68, 57.8 , 85.3 ,
       83.67, 75.69, 75.9 , 55.02, 77.5 , 88.83, 75.5 , 92.3 , 67.2 ,
       62.3 , 72.48, 59.2 , 72.8 , 60.13, 94.3 , 89.8 , 81.3 , 82.02,
       76.89, 60.16, 77.6 , 69.33, 87.5 , 74.5 , 46.  , 74.88, 68.33,
       84.67, 87.2 , 66.3 , 58.6 , 85.36, 68.7 , 55.54, 68.89, 82.3 ,
       58.2 , 48.34, 54.2 , 92.7 , 71.01, 53.8 , 56.6 , 66.77, 71.6 ,
       90.6 , 82.66, 89.08, 93.6 , 72.3 , 69.7 , 84.17, 92.1 , 90.4 ,
```

92.67, 64.33, 62.81, 67.17, 95.1 , 68.83, 97.8 , 88.1 , 48.8 ,
60.05, 69.07, 89.3 , 86.3 , 94.6 , 86.25, 79.23, 97. , 84.75,
89.2 , 77.54, 60.8 , 84.6 , 86.2 , 65.16, 94.33, 74.67, 94.7 ,
92.2 , 90.33, 73.61, 53.6 , 58.4 , 78.86, 76.44, 78.66, 97.5 ,
62.26, 69.84, 78.13, 60.83, 51.3 , 57.67, 67.9 , 56.3 , 88.6 ,
52.5 , 76.33, 94. , 64.88, 80.75, 71.66, 50.3 , 76.67, 67.57,
64.45, 69.5 , 72.15, 59.77, 61.4 , 72.1 , 90.25, 97.4 , 89.4 ,
57.5 , 70.8 , 56.4 , 96.2 , 88.77, 96. , 83.2 , 69.17, 80.33,
79.04, 59.8 , 81.66, 72.46, 92.5 , 50.5 , 96.8 , 93.5 , 65.33,
92.4 , 72.66, 87.33, 94.8 , 60.3 , 90.3 , 61.17, 82.25, 88.5 ,
75.11, 70.33, 85.8 , 88.8 , 45.6 , 58.33, 76.24, 45. , 68.3 ,
89.75, 69.67, 88.2 , 65.56, 74.12, 86.54, 57.11, 65.66, 85.67,
96.3 , 72.31, 70.3 , 77.86, 94.75, 69.32, 84.9 , 60.1 , 86.6 ,
70.15, 66.62, 79.16, 76.7 , 65.25, 93.3 , 68.15, 80.25, 76.77,
65.9 , 91.25, 54.14, 55.55, 95.65, 75.91, 66.2 , 70.66, 59.4 ,
58.92, 51. , 63.7 , 85.5 , 70.83, 81.7 , 63.8 , 75.25, 43. ,
61.16, 56.12, 60.66, 69.58, 70.04, 79.67, 49.67, 76.3 , 75.67,
61.3 , 82.33, 88.33, 88.45, 96.75, 84.25, 78.5 , 77.85, 82.46,
74.3 , 73.67, 60.01, 91.33, 69.16, 91.7 , 65.83, 74.83, 58.5 ,
72.17, 87.6 , 75.85, 79.87, 85.9 , 85.6 , 85.7 , 73.11, 88.66,
82.56, 78.34, 89.33, 66.15, 87.25, 69.53, 53.17, 79.19, 77.16,
54.08, 52.2 , 83.5 , 85.57, 60.33, 67.23, 47.6 , 73.25, 69.88,
84.53, 89.9 , 79.3 , 93.1 , 92.25, 91.58, 75.14, 73.3 , 77.1 ,
79.5 , 59.6 , 50.4 , 74.89, 63.33, 77.23, 83.43, 83.3 , 95.7 ,
90.8 , 75.75, 47.2 , 86.31, 85.17, 63.83, 64.57, 73.69, 66.86,
81.75, 55.6 , 62.12, 80.58, 74.96, 70.14, 69.54, 61.83, 72.33,
74.33, 89.04, 71.67, 60.36, 90.66, 60.5 , 70.26, 57.2 , 87.69,
85.56, 64.2 , 45.5 , 70.5 , 93.9 , 57.58, 85.1 , 64.1 , 79.1 ,
91.83, 88.93, 46.33, 94.17, 95.9 , 81.9 , 76.12, 84.28, 55.4 ,
58.04, 91.8 , 74.7 , 83.25, 91.4 , 54.66, 82.9 , 66.46, 71.52,
56.17, 59.66, 62.7 , 78.25, 61.92, 68.16, 53.83, 47.83, 92.75,
63.44, 87.05, 92.08, 68.69, 55.33, 89.67, 63.77, 74.53, 60.6 ,
77.83, 56.04, 70.67, 65.1 , 87.3 , 56.22, 96.4 , 83.62, 74.44,
97.9 , 57.33, 71.17, 95.08, 84.2 , 97.6 , 57.83, 93.41, 79.33,
64.83, 75.66, 75.8 , 62.33, 92.9 , 92.8 , 70.92, 81.42, 90.9 ,
52.7 , 82.03, 73.63, 75.1 , 87.17, 73.33, 62.23, 61.33, 52.9 ,
83.88, 83.75, 83.83, 67.74, 59.38, 68.04, 77.81, 89.26, 74.71,
53.2 , 53.85, 65.54, 51.6 , 68.77, 86.29, 88.75, 49.6 , 77.33,
58.8 , 71.85, 66.17, 53.44, 66.83, 93.2 , 85.33, 72.9 , 57.1 ,
98.7 , 88.16, 54.33, 85.42, 90.5 , 74.16, 98.2 , 82.71, 90.03,
85.23, 63.16, 96.33, 84.09, 74.05, 81.16, 40. , 60.06, 89.16,
69.89, 61.23, 79.82, 65.72, 55.8 , 76.85, 69.3 , 90.17, 53.16,
61.85, 89.58, 83.23, 67.3 , 93.33, 52.8 , 73.17, 52.34, 63.5 ,
77.11, 62.11, 97.2 , 78.88, 65.12, 55.44, 53.55, 80.22, 62.9 ,
55.66, 94.91, 76.83, 64.08, 83.34, 85.88, 63.3 , 53.4 , 54.8 ,
54.5 , 74.25, 79.75, 60.44, 85.66, 55.5 , 51.23, 90.83, 67.1 ,
83.16, 79.7 , 56.1 , 68.17, 82.53, 87.14, 63.9 , 74.14, 89.91,

```
       77.06, 60.42, 75.33, 96.25, 69.12, 77.56, 86.46, 81.26, 64.31,
       71.83, 86.91, 81.25, 54.83, 59.11, 91.1 , 81.67, 53.33, 82.55])
```

[23]:
```
df['12board'].unique()
```

[23]:
```
array(['board of intermediate education,ap', 'cbse', 'state board',
       'mp board', 'isc', 'icse', 'karnataka pre university board', 'up',
       'p u board, karnataka', 'dept of pre-university education', 'bie',
       'kerala state hse board', 'up board', '0', 'bseb', 'chse', 'puc',
       ' upboard',
       'state  board of intermediate education, andhra pradesh',
       'karnataka state board',
       'west bengal state council of technical education', 'wbchse',
       'maharashtra state board', 'ssc', 'isc board',
       'sda matric higher secondary school', 'uttar pradesh board', 'ibe',
       'chsc', 'board of intermediate', 'isce', 'upboard', 'sbtet',
       'hisher seconadry examination(state board)', 'pre university',
       'borad of intermediate', 'j & k board',
       'intermediate board of andhra pardesh', 'rbse',
       'central board of secondary education', 'jkbose', 'hbse',
       'board of intermediate education', 'state', 'ms board', 'pue',
       'intermediate state board', 'stateboard', 'hsc',
       'electonincs and communication(dote)', 'karnataka pu board',
       'government polytechnic mumbai , mumbai board', 'pu board',
       'baord of intermediate education', 'apbie', 'andhra board',
       'tamilnadu stateboard',
       'west bengal council of higher secondary education',
       'cbse,new delhi', 'u p board', 'intermediate', 'biec,patna',
       'diploma in engg (e &tc) tilak maharashtra vidayapeeth',
       'hsc pune', 'pu board karnataka', 'kerala', 'gsheb',
       'up(allahabad)', 'nagpur', 'st joseph hr sec school',
       'pre university board', 'ipe', 'maharashtra', 'kea', 'apsb',
       'himachal pradesh board of school education', 'staae board',
       'international baccalaureate (ib) diploma', 'nios',
       'karnataka board of university',
       'board of secondary education rajasthan', 'uttarakhand board',
       'ua', 'scte vt orissa', 'matriculation',
       'department of pre-university education', 'wbscte',
       'preuniversity board(karnataka)', 'jharkhand accademic council',
       'bieap', 'msbte (diploma in computer technology)',
       'jharkhand acamedic council (ranchi)',
       'department of pre-university eduction', 'biec', 'all india board',
       'sjrcw', ' board of intermediate', 'msbte',
       'sri sankara vidyalaya', 'chse, odisha', 'bihar board',
       'maharashtra state(latur board)', 'rajasthan board', 'mpboard',
       'state board of technical eduction panchkula', 'upbhsie', 'apbsc',
       'state board of technical education and training',
```

'secondary board of rajasthan',
'tamilnadu higher secondary education board',
'jharkhand academic council',
'board of intermediate education,hyderabad', 'up baord', 'pu',
'dte', 'board of secondary education', 'pre-university',
'board of intermediate education,andhra pradesh',
'up board , allahabad', 'srv girls higher sec school,rasipuram',
'intermediate board of education,andhra pradesh',
'intermediate board examination',
'department of pre-university education, bangalore',
'stmiras college for girls', 'mbose',
'department of pre-university education(government of karnataka)',
'dpue', 'msbte pune', 'board of school education harayana',
'sbte, jharkhand', 'bihar intermediate education council, patna',
'higher secondary', 's j polytechnic', 'latur',
'board of secondary education, rajasthan', 'jyoti nivas', 'pseb',
'biec-patna', 'board of intermediate education,andra pradesh',
'chse,orissa', 'pre-university board', 'mp', 'intermediate board',
'govt of karnataka department of pre-university education',
'karnataka education board',
'board of secondary school of education', 'pu board ,karnataka',
'karnataka secondary education board', 'karnataka sslc',
'board of intermediate ap', 'u p', 'state board of karnataka',
'directorate of technical education,banglore', 'matric board',
'andhpradesh board of intermediate education',
'stjoseph of cluny matrhrsecschool,neyveli,cuddalore district',
'bte up', 'scte and vt ,orissa', 'hbsc',
'jawahar higher secondary school', 'nagpur board', 'bsemp',
'board of intermediate education, andhra pradesh',
'board of higher secondary orissa',
'board of secondary education,rajasthan(rbse)',
'board of intermediate education:ap,hyderabad', 'science college',
'karnatak pu board', 'aissce', 'pre university board of karnataka',
'bihar', 'kerala state board', 'uo board', 'cicse',
'karnataka board', 'tn state board',
'kolhapur divisional board, maharashtra',
'jaycee matriculation school',
'board of higher secondary examination, kerala',
'uttaranchal state board', 'intermidiate', 'bciec,patna', 'bice',
'karnataka state', 'state broad', 'wbbhse', 'gseb',
'uttar pradesh', 'ghseb', 'board of school education uttarakhand',
'gseb/technical education board', 'msbshse,pune',
'tamilnadu state board', 'board of technical education',
'kerala university', 'uttaranchal shiksha avam pariksha parishad',
'chse(concil of higher secondary education)',
'bright way college, (up board)', 'board of intermidiate',
'higher secondary state certificate', 'karanataka secondary board',

'maharashtra board', 'andhra pradesh state board', 'cgbse',
'diploma in computers', 'bte,delhi', 'rajasthan board ajmer',
'mpbse', 'pune board', 'state board of technical education',
'gshseb', 'amravati divisional board',
'dote (diploma - computer engg)', 'up bord',
'karnataka pre-university board', 'jharkhand board',
'punjab state board of technical education & industrial training',
'department of technical education',
'sri chaitanya junior kalasala', 'state board (jac, ranchi)',
'gujarat board', 'aligarh muslim university',
'tamil nadu state board', 'hse', 'karnataka secondary education',
'state board ', 'karnataka pre unversity board',
'ks rangasamy institute of technology',
'karnataka board secondary education', 'narayana junior college',
'bteup', 'board of intermediate(bie)', 'hsc maharashtra board',
'tamil nadu state', 'uttrakhand board', 'psbte',
'stateboard/tamil nadu', 'intermediate council patna',
'technical board, punchkula', 'board of intermidiate examination',
'sri kannika parameswari highier secondary school, udumalpet',
'ap board', 'nashik board', 'himachal pradesh board',
'maharashtra satate board',
'andhra pradesh board of secondary education',
'tamil nadu polytechnic',
'maharashtra state board mumbai divisional board',
'department of pre university education',
'dav public school,hehal', 'board of intermediate education, ap',
'rajasthan board of secondary education',
'department of technical education, bangalore', 'chse,odisha',
'maharashtra nasik board',
'west bengal council of higher secondary examination (wbchse)',
'holy cross matriculation hr sec school', 'cbsc',
'pu board karnataka', 'biec patna', 'kolhapur', 'bseb, patna',
'up board allahabad', 'intermideate', 'nagpur board,nagpur',
'diploma(msbte)', 'dav public school',
'pre university board, karnataka', 'ssm srsecschool', 'state bord',
'jstb,jharkhand', 'intermediate board of education',
'mp board bhopal', 'pub', 'madhya pradesh board',
'bihar intermediate education council',
'west bengal council of higher secondary eucation',
'isc board , new delhi', 'mpc',
'certificate for higher secondary education (chse)orissa',
'maharashtra state board for hsc',
'board of intermeadiate education', 'latur board',
'andhra pradesh', 'karnataka pre-university',
'lucknow public college', 'nagpur divisional board',
'ap intermediate board', 'cgbse raipur', 'uttranchal board',
'jiec', 'central board of secondary education, new delhi',

```
       'bihar school examination board patna',
       'state board of technical education harayana', 'mp-bse',
       'up bourd', 'dav public school sec 14',
       'haryana state board of technical education chandigarh',
       'council for indian school certificate examination',
       'jaswant modern school', 'madhya pradesh open school',
       'aurangabad board', 'j&k state board of school education',
       'diploma ( maharashtra state board of technical education)',
       'board of technicaleducation ,delhi',
       'maharashtra state boar of secondary and higher secondary education',
       'hslc (tamil nadu state board)',
       'karnataka state examination board', 'puboard', 'nasik',
       'west bengal board of higher secondary education',
       'up board,allahabad', 'board of intrmediate education,ap', 'cbese',
       'karnataka state pre- university board',
       'state board - west bengal council of higher secondary education :
wbchse',
       'maharashtra state board of secondary & higher secondary education',
       'biec, patna', 'state syllabus', 'cbse board', 'scte&vt',
       'board of intermediate,ap',
       'secnior secondary education board of rajasthan',
       'maharashtra board, pune', 'rbse (state board)',
       'board of intermidiate education,ap',
       'board of high school and intermediate education uttarpradesh',
       'higher secondary education',
       'board fo intermediate education, ap', 'intermedite',
       'ap board for intermediate education', 'ahsec',
       'punjab state board of technical education & industrial training,
chandigarh',
       'state board - tamilnadu', 'jharkhand acedemic council',
       'scte & vt (diploma)', 'karnataka pu',
       'board of intmediate education ap', 'up-board',
       'boardofintermediate'], dtype=object)
```

[24]:
```
df['CollegeID'].unique()
```

[24]:
```
array([1141, 5807,   64, …, 3572, 6327, 4883])
```

[25]:
```
df['CollegeTier'].unique()
```

[25]:
```
array([2, 1])
```

[26]:
```
df['Degree'].unique()
```

[26]:
```
array(['B.Tech/B.E.', 'MCA', 'M.Tech./M.E.', 'M.Sc. (Tech.)'],
      dtype=object)
```

[27]:
```
df['Specialization'].unique()
```

[27]:
```
array(['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'],
      dtype=object)
```

[28]:
```
df['collegeGPA'].unique()
```

[28]:
```
array([78.  , 70.06, 70.  , …, 65.05, 74.73, 70.42])
```

[29]:
```
df['CollegeCityID'].unique()
```

[29]:
```
array([1141, 5807,   64, …, 3572, 6327, 4883])
```

[30]:
```
df['CollegeCityTier'].unique()
```

[30]:
```
array([0, 1])
```

[31]:
```
df['CollegeState'].unique()
```

[31]:
```
array(['Andhra Pradesh', 'Madhya Pradesh', 'Uttar Pradesh', 'Delhi',
       'Karnataka', 'Tamil Nadu', 'West Bengal', 'Maharashtra', 'Haryana',
       'Telangana', 'Orissa', 'Punjab', 'Kerala', 'Gujarat', 'Rajasthan',
       'Chhattisgarh', 'Uttarakhand', 'Jammu and Kashmir', 'Jharkhand',
       'Himachal Pradesh', 'Bihar', 'Assam', 'Goa', 'Sikkim',
       'Union Territory', 'Meghalaya'], dtype=object)
```

[32]:
```
df['GraduationYear'].unique()
```

[32]:
```
array([2011, 2012, 2014, 2016, 2013, 2010, 2015, 2009, 2017,    0, 2007])
```

[33]:
```
df['English'].unique()
```

[33]:
```
array([515, 695, 615, 635, 545, 560, 590, 605, 565, 495, 380, 395, 485,
       685, 465, 455, 385, 370, 625, 575, 415, 535, 580, 475, 570, 430,
       450, 510, 425, 555, 300, 505, 440, 525, 420, 640, 444, 630, 665,
       675, 325, 405, 375, 315, 710, 345, 250, 350, 275, 360, 265, 595,
       585, 520, 500, 735, 765, 335, 490, 660, 355, 530, 365, 655, 730,
       445, 720, 645, 650, 875, 534, 454, 544, 295, 285, 435, 464, 705,
       554, 745, 280, 825, 290, 715, 310, 215, 700, 870, 305, 524, 755,
       790, 800, 205, 725, 780, 404, 770, 805, 180, 830, 795, 255, 324,
       775, 394, 240, 225, 850, 684, 334])
```

[34]:
```
df['Logical'].unique()
```

[34]:
```
array([585, 610, 545, 625, 555, 435, 670, 565, 455, 605, 580, 425, 520,
       530, 495, 445, 535, 360, 335, 510, 570, 375, 405, 485, 475, 525,
       640, 595, 560, 340, 395, 415, 465, 505, 385, 460, 410, 500, 645,
       480, 355, 450, 440, 470, 255, 305, 590, 630, 365, 350, 325, 400,
       205, 655, 295, 345, 390, 665, 515, 540, 680, 245, 620, 420, 575,
       635, 554, 315, 615, 215, 370, 300, 274, 685, 324, 675, 650, 464,
       684, 275, 334, 544, 454, 534, 404, 795, 285, 715, 700, 674, 690,
       695, 394, 270, 705, 310, 490, 330, 280, 735, 380, 290, 265, 240,
       195, 235, 660])
```

[35]:
```
df['Quant'].unique()
```

[35]:
```
array([525, 780, 370, 625, 465, 620, 380, 590, 530, 545, 565, 715, 470,
       645, 355, 515, 435, 445, 485, 270, 630, 575, 405, 605, 385, 695,
       450, 295, 430, 415, 635, 475, 460, 825, 500, 455, 554, 595, 495,
       665, 250, 310, 325, 390, 510, 535, 340, 440, 705, 534, 400, 395,
       570, 750, 330, 320, 454, 365, 615, 505, 425, 235, 210, 585, 810,
       555, 735, 560, 524, 690, 870, 765, 675, 520, 655, 305, 725, 840,
       650, 375, 720, 265, 280, 464, 404, 800, 680, 260, 674, 760, 345,
       335, 165, 685, 544, 215, 180, 795, 200, 860, 334, 285, 514, 195,
       494, 214, 275, 315, 324, 175, 684, 225, 740, 805, 444, 410, 135,
       255, 220, 755, 855, 145, 245, 885, 120, 900, 794, 775, 745, 504,
```

```
       820, 150, 710, 190, 185, 155, 580, 394])
```

[36]:
```
df['Domain'].unique()
```

[36]:
```
array([ 0.63597876,  0.96060325,  0.45087658,  0.97439611,  0.12450207,
       -1.        ,  0.35653649,  0.8295846 ,  0.69447933,  0.49359639,
        0.76567358,  0.9682375 ,  0.22948175,  0.53838689,  0.30840058,
        0.91139528,  0.56326782,  0.86468541,  0.64938971,  0.74475835,
        0.88412251,  0.88162007,  0.20739217,  0.48674701,  0.67074315,
        0.62264292,  0.41383826,  0.52592258,  0.73579571,  0.13044174,
        0.23780284,  0.11213944,  0.37755142,  0.06696071,  0.08005528,
        0.92564577,  0.84312373,  0.91686996,  0.78330354,  0.60005718,
        0.79293628,  0.79358061,  0.16563309,  0.75537512,  0.99990456,
        0.33878635,  0.91077016,  0.98205712,  0.84224832,  0.01854094,
        0.05316031,  0.94211655,  0.12301673,  0.48834798,  0.37605959,
        0.0587928 ,  0.10487136,  0.60064396,  0.70409041,  0.14478989,
        0.81941653,  0.65576694,  0.02106623,  0.44461772,  0.83762073,
        0.72598415,  0.95389978,  0.04099931,  0.02196911,  0.3423149 ,
        0.53586282,  0.90148957,  0.96177212,  0.67964464,  0.93839914,
        0.19015341,  0.99546472,  0.99000876,  0.97879929,  0.41433743,
        0.8246664 ,  0.49063696,  0.96600692,  0.43696265,  0.99225919,
        0.97629256,  0.98546139,  0.90915194,  0.86372418,  0.07454627,
        0.66183448,  0.93037061,  0.95224557,  0.96090309,  0.97706647,
        0.55738951,  0.7908818 ,  0.99966434,  0.8799152 ,  0.85882669,
        0.97166349,  0.19376844,  0.24545566,  0.98374997,  0.98720709,
        0.97952174,  0.86873659,  0.45901584,  0.02231329,  0.94513486,
        0.27604723,  0.53678267,  0.4239514 ,  0.29876913,  0.20671062,
        0.12569018,  0.16363093,  0.99138693,  0.94327216,  0.60703404,
        0.99139758,  0.25577819,  0.96221701,  0.99123063,  0.87550391,
        0.02331217,  0.44714789,  0.14325654,  0.84084097,  0.33618507,
        0.30451091,  0.27925851,  0.84980271,  0.03293702,  0.65410747,
        0.28677781,  0.21678549,  0.95868202,  0.32874638,  0.98466189,
        0.99674449,  0.99434211,  0.70082616,  0.99887611,  0.86262547,
        0.60553355,  0.68456507,  0.71452936,  0.44619825,  0.92514031,
        0.99526571,  0.59828069,  0.98652502,  0.11166096,  0.11655256,
        0.8957765 ,  0.1559085 ,  0.99764277,  0.88790332,  0.88570368,
        0.99405087,  0.02781507,  0.90394078,  0.93174969,  0.09575376,
        0.1925871 ,  0.99868025,  0.03114969,  0.61061199,  0.75837988,
        0.63958738,  0.27350033,  0.0109953 ,  0.21325143,  0.99925029,
        0.06844575,  0.04222313,  0.77657832,  0.75618011,  0.99808656,
        0.99982917,  0.01656464,  0.00815478,  0.89984384,  0.92879365,
        0.99769826,  0.9956138 ,  0.9788553 ,  0.03056599,  0.04219243,
        0.18477158,  0.92460994,  0.81062051,  0.99742783,  0.66033499,
        0.99615597,  0.99859145,  0.00275015,  0.51986392,  0.99729897,
        0.20273159,  0.38009183,  0.52173598,  0.0139623 ,  0.24249992,
        0.29429542,  0.4099522 ,  0.80558321,  0.24320866,  0.96532716,
        0.4845907 ,  0.97524659,  0.1798739 ,  0.37113867,  0.86613982,
```

```
       0.11502273,  0.99991041,  0.99179243,  0.22105882,  0.0417332 ,
       0.06222129,  0.83968608,  0.41552456,  0.25302837,  0.99657055,
       0.97289871,  0.99896694,  0.99391731,  0.52911566,  0.55321638,
       0.01770484,  0.91541843,  0.2995973 ,  0.66732645,  0.27845741,
       0.00853725,  0.99858824,  0.28278814,  0.08874741,  0.79984821,
       0.99653553,  0.90474069,  0.93858826])
```

[37]:
```
df['ComputerProgramming'].unique()
```

[37]:
```
array([445,  -1, 395, 615, 645, 405, 735, 385, 485, 605, 495, 355, 515,
       545, 425, 525, 455, 475, 565, 535, 335, 345, 465, 415, 435, 155,
       375, 555, 305, 315, 804, 285, 575, 505, 195, 225, 595, 275, 334,
       365, 685, 655, 625, 585, 665, 325, 235, 255, 205, 494, 695, 635,
       215, 464, 295, 394, 245, 715, 265, 135, 105, 524, 165, 175, 125,
       675, 454, 745, 185, 214, 145, 544, 725, 840, 404, 755, 705, 115,
       554])
```

[38]:
```
df['ElectronicsAndSemicon'].unique()
```

[38]:
```
array([ -1, 466, 233, 366, 324, 266, 333, 356, 420, 260, 228, 388, 300,
       292, 433, 196, 200, 164, 400, 484, 500, 452, 516, 166, 533, 566,
       612, 133, 548])
```

[39]:
```
df['ComputerScience'].unique()
```

[39]:
```
array([ -1, 407, 346, 376, 500, 438, 315, 253, 469, 192, 530, 284, 223,
       561, 684, 592, 623, 653, 130, 715])
```

[40]:
```
df['MechanicalEngg'].unique()
```

[40]:
```
array([ -1, 469, 313, 286, 253, 366, 446, 206, 438, 332, 393, 383, 260,
       561, 553, 376, 526, 284, 409, 473, 340, 223, 420, 538, 346, 435,
       512, 407, 580, 280, 358, 500, 315, 254, 616, 564, 233, 306, 461,
       180, 606, 623])
```

[41]:
```
df['ElectricalEngg'].unique()
```

[41]:
```
array([ -1, 484, 606, 393, 500, 553, 580, 446, 420, 324, 388, 356, 313,
       633, 516, 366, 612, 452, 526, 548, 228, 433, 473, 676, 292, 660,
       411, 286, 340, 260, 206])
```

[42]:
```
df['CivilEngg'].unique()
```

[42]:
```
array([ -1, 320, 400, 388, 260, 440, 356, 292, 500, 200, 300, 452, 322,
       340, 166, 277, 516, 380, 433, 280, 420, 460, 480])
```

[43]:
```
df['conscientiousness'].unique()
```

[43]:
```
array([ 0.9737, -0.7335,  0.2718,  0.0464, -0.881 , -0.3027,  1.7081,
       -0.0154, -0.159 , -1.308 , -2.272 ,  0.1282,  0.3555,  0.7027,
        1.7465,  1.1336,  0.8463,  0.8192, -0.1082, -1.0355, -0.4463,
        0.4155,  0.99  , -3.1994, -0.4173,  1.5644, -0.4854, -1.0208,
        0.3941, -0.8772,  0.51  , -0.5899, -2.5039,  1.2828,  0.335 ,
       -0.3014,  1.8517, -1.1644, -2.2351,  0.6646, -0.2628, -1.8825,
       -1.4517,  0.5591,  1.4208, -0.7264, -0.5116, -1.7389,  0.2009,
       -0.0696, -2.5811, -2.3134,  1.2772, -2.8879,  1.4374, -1.3447,
        0.1623,  1.7156, -1.9629, -2.457 ,  1.9953, -2.0262, -2.1175,
       -2.7443, -1.4606,  0.8578, -1.1901, -0.7651, -0.5719, -2.1698,
       -1.8083,  1.592 , -0.9969, -1.3742, -1.4992, -1.5953, -3.6631,
        1.1283, -3.606 ,  0.1788,  0.2782, -1.6538, -3.3539, -1.1128,
       -3.3188,  0.4285,  0.7419, -0.6491, -0.51  , -0.5236, -0.9653,
       -3.1752,  0.7208,  1.3215, -1.6924, -0.1855, -0.6749,  0.2318,
        1.5533,  1.0768,  1.3686, -0.5332, -0.2632, -1.2287, -1.5765,
        1.9011,  1.0896,  0.215 , -3.4624,  0.3836, -3.0448, -2.6007,
        1.2056, -1.295 , -2.7357, -0.0415,  1.6692,  0.626 , -2.4266,
       -2.1561, -2.8903, -4.1267,  0.4034, -1.9243, -1.3025, -1.5964,
        1.7852,  0.6696,  0.5522,  0.8479, -1.0135, -0.1982, -3.8933,
       -3.5085, -3.7496, -4.0369, -1.977 , -3.0315,  1.2266, -0.4595,
        0.8986])
```

[44]:
```
df['agreeableness'].unique()
```

[44]:
```
array([ 0.8128,  0.3789,  1.7109,  0.3448, -0.2793, -0.6201, -0.1054,
        1.2114,  0.5454,  1.1248,  0.0328,  0.7119,  1.9048,  1.0449,
        0.2668,  0.9688, -0.5913, -2.1186,  0.8027,  1.2028,  0.1888,
        1.3779, -1.8393,  0.6568, -0.4536, -0.5213,  0.2124,  1.2808,
       -1.1196,  0.2578, -2.4516,  1.7488, -0.1206,  0.0924, -0.0842,
       -0.4353,  1.5444, -0.9531, -2.6847, -0.1232, -3.7836,  1.4368,
        0.8784, -1.4526,  0.0459, -0.2871,  1.0858,  1.7878, -0.7866,
        0.8229, -1.6191, -1.2861,  0.6178, -0.2012,  0.5008, -0.9033,
        1.5538, -4.2831,  0.8518,  0.1498, -1.9953, -1.3713, -2.9314,
       -1.2153, -0.7473, -0.5523, -1.0593,  0.4934, -1.7856, -1.9521,
       -5.6151,  1.5928, -0.6693, -1.8855, -2.4633, -2.6193,  0.7348,
       -0.8865, -1.6833,  1.3198, -5.1156, -0.3183,  0.3731, -1.2543,
       -0.7993, -2.1903,  0.7816,  0.6009,  0.3002, -3.6171, -2.6181,
       -1.5273, -2.7754,  1.5081, -2.1513, -2.2851,  1.6708, -0.7863,
       -2.3073, -3.4506, -0.3684,  0.7135, -0.6867, -3.0874, -5.7816,
        0.3838,  0.3123,  0.1125, -1.4859, -0.0873, -3.1176,  0.4488,
        0.8993, -3.1264, -3.0094, -2.0733, -3.9501, -2.9511, -1.7223,
       -0.1374, -4.7826, -1.0905, -0.6504,  0.0875, -3.8284, -1.4883,
        0.9117,  1.5293,  0.6211, -2.7846, -0.1334,  0.4395, -1.0203,
        0.9028, -1.7056, -3.2434,  0.5121,  0.8351,  0.0762, -1.6313,
       -2.4243, -1.1373, -3.3994, -2.6583, -0.9884,  1.3476, -0.4778,
       -0.0651, -0.832 ])
```

[45]:
```
df['extraversion'].unique()
```

[45]:
```
array([ 0.5269,  1.2396,  0.1637, -0.344 , -1.0697, -2.2954, -1.0379,
        0.01  , -0.6048, -0.9122,  0.0914,  0.8171, -0.7585, -0.598 ,
        0.672 ,  0.7785, -1.0659,  1.3933, -0.2714, -1.3599, -1.9881,
        0.1357, -0.9245, -1.7954,  0.0552, -0.0537,  1.0859,  0.3174,
        2.1129,  0.4711,  0.6248,  0.2366, -0.5349, -0.4511, -0.6343,
       -0.7794,  0.3817,  1.8331, -2.2308, -0.6582, -0.2974, -2.6028,
       -2.4491, -1.2196,  2.1617, -0.1988, -0.4891,  0.9322, -1.2148,
        1.7007,  1.1437, -0.1437,  1.8543,  1.547 ,  0.8809, -1.6807,
        0.7083,  0.5994,  1.1074, -1.5776,  0.9623, -1.8344, -1.5051,
       -0.1626,  0.926 , -3.2176, -1.3733,  1.2525, -1.4688, -2.1418,
        1.688 ,  1.1558, -1.6502,  0.4906,  1.5428, -1.527 ,  1.9782,
       -1.9405,  0.2113, -2.3759,  0.2075, -2.0856, -0.3803,  0.2729,
       -4.6009, -1.2511, -0.0319,  2.3154, -3.525 , -0.8157, -0.6355,
        0.6984,  1.3977, -3.0639,  2.1234,  2.008 , -2.3396, -1.9042,
        1.1804, -2.775 ,  0.73  ,  0.164 , -0.824 , -2.1219, -0.7068,
        1.4702,  1.4267, -1.1422, -1.6865, -0.2882, -2.6662,  1.0348,
       -1.0334, -2.9565, -4.2935, -2.7565,  0.065 , -0.1996, -3.537 ,
       -2.0131, -0.4226, -3.8636,  1.5791, -0.1408, -1.9408, -0.7026,
       -2.4485, -2.9102, -0.6339,  1.6484, -3.3713,  0.3034,  1.3614,
        2.5354, -0.0933,  1.9801,  0.3292, -0.8703, -1.0116, -0.4899,
        0.1138, -1.7086,  0.6388, -3.9861, -2.521 , -2.8113,  0.9042,
        0.2477, -3.8324, -0.3149, -0.2516, -4.4472, -1.7083, -1.2056])
```

[46]:
```
df['nueroticism'].unique()
```

[46]:
```
array([ 1.3549 , -0.1076 , -0.8682 , -0.4078 ,  0.09163, -0.7415 ,
       -2.0092 ,  0.1459 ,  0.9066 ,  0.1798 , -0.995  , -0.2902 ,
       -0.6147 , -1.6289 , -0.2344 ,  0.06223,  0.7798 , -0.4879 ,
        0.5323 ,  1.8249 , -0.3612 ,  0.0623 , -1.2303 , -1.5021 ,
        1.1601 , -1.8824 , -0.735  ,  0.2727 , -2.1998 , -1.2486 ,
       -1.1218 ,  0.2973 ,  1.7074 ,  0.26793, -2.2879 ,  1.0024 ,
        0.653  , -0.4821 , -0.6428 , -0.349  ,  1.5404 ,  0.3995 ,
        0.6498 ,  1.794  ,  0.0192 ,  1.0333 ,  0.4148 ,  0.8848 ,
       -0.7603 ,  0.88483, -0.5253 ,  1.2869 ,  1.4724 , -0.8778 ,
       -0.1727 ,  0.0035 , -1.1128 ,  1.6672 ,  0.64983,  1.1199 ,
       -2.3895 , -1.7556 , -2.136  ,  0.53233, -0.0552 , -0.26087,
        0.17983, -0.6134 ,  0.219  ,  0.5262 ,  0.76733,  0.00353,
       -1.3753 , -0.29027,  1.0611 , -0.7015 ,  2.6475 ,  0.7673 ,
        1.4136 ,  0.29733,  0.0917 , -1.4653 ,  0.7967 ,  2.2949 ,
       -1.3478 , -0.2609 ,  1.5899 , -0.05527,  2.1774 ,  1.4297 ,
       -0.3414 , -0.78967, -1.1422 ,  2.301  , -0.9953 ,  0.9169 ,
       -0.52527, -0.87777, -2.643  , -0.40777, -0.01   , -0.4371 ,
        1.58993,  2.4278 ,  1.3255 ,  2.9349 ,  0.6204 , -2.0529 ,
       -1.8179 , -1.05407,  2.0599 ,  2.0475 , -1.3184 ,  1.85433,
       -0.5644 , -1.11277, -0.365  , -1.4066 ,  1.2374 , -1.9033 ,
```

```
 1.9424 , -2.5163 ,  0.70853, -1.5828 ,  1.67803,  1.7662 ,
-1.58287,  2.53   , -0.70157,  0.44423,  0.4442 ,  2.1743 ,
 1.9207 ,  0.3561 , -1.7004 , -2.5047 ,  1.1492 ,  0.3756 ,
-1.671  ,  0.8457 , -0.17277, -0.99527, -0.76027, -1.34787,
-1.23027,  0.7086 ,  1.5018 , -1.3255 ,  0.973  , -2.1704 ,
 1.06113,  2.1187 ,  0.40413, -0.0846 , -1.9354 , -0.43717,
 1.41363,  3.3525 ,  0.4041 ,  1.76613,  1.14923, -2.2627 ,
-0.64277, -1.46537,  1.32553,  0.1477 , -0.7897 , -0.61337,
 1.11983,  3.235  ,  0.2679 ,  1.70743, -0.34897,  2.4125 ,
 1.35483,  0.62043,  2.6814 , -1.0541 ,  1.8543 , -1.1911 ,
 0.41483, -0.9659 ,  3.0617 ,  1.23733, -0.7496 ,  2.765  ,
 1.82493,  2.7356 , -1.49467,  2.4712 , -0.5382 ,  2.29493,
-0.5958 ,  2.47123, -1.5899 , -0.08457, -1.7591 ,  3.3152 ,
 0.35603,  2.2068 , -1.3008 , -0.96597, -0.8177 ,  0.7493 ,
 1.678  ,  1.00233,  1.50173,  0.97293,  1.0747 ,  1.5578 ,
 2.5546 , -1.14217,  0.6605 ,  2.5593 ,  0.9553 ,  0.2759 ,
 2.0306 ])
```

[47]:
```
df['openess_to_experience'].unique()
```

[47]:
```
array([-0.4455,  0.8637,  0.6721, -0.9194, -0.1295, -0.8608, -1.0872,
        1.247 , -0.2859,  0.0973,  0.0284, -1.2354,  1.2528,  1.4386,
        0.3444, -1.3539, -2.7769, -5.0763, -0.6692, -0.2875,  1.1343,
       -0.0943, -0.7615,  0.2889, -0.4776,  1.0554,  1.8224,  0.6603,
       -1.4356, -1.359 , -3.1602,  0.1864,  0.5024, -1.244 , -0.1543,
       -0.6035, -5.477 , -1.8189, -2.3937,  0.3049,  0.8183,  0.4805,
        1.6302, -2.2021,  1.2923,  0.9763, -2.9731, -1.0524, -1.6273,
        1.6082, -5.2679, -0.169 , -1.0774, -3.9605, -0.0506, -0.5245,
       -1.8673, -0.8799, -0.9984,  0.5419, -4.5015, -2.1833, -2.3415,
        0.1275, -0.643 , -1.3934,  1.4502, -6.9925, -1.7093, -0.8782,
       -3.4471, -2.0105, -0.4137,  0.4234, -1.5513, -1.1169, -1.425 ,
        0.8973,  0.0916, -2.5853, -2.0253,  0.0679,  0.7788, -0.4139,
       -3.3518, -3.735 , -2.3412, -2.9686,  1.0031, -4.3099, -0.0844,
        1.0158, -2.7595,  1.3976, -0.406 ,  1.4186, -0.4601, -1.0458,
       -5.8428, -2.0648,  0.167 ,  0.7941,  1.2121, -3.5434, -6.8009,
        0.1187,  0.585 , -3.9266, -2.6572, -3.6051, -3.763 , -5.686 ,
       -1.1291, -0.0167, -2.8152, -1.8278, -5.6512,  0.7631, -5.4595,
        1.0395, -0.4392, -3.1311, -0.5081, -1.4724, -2.3017, -1.6662,
       -1.9463,  0.9404, -2.4202, -0.2511,  0.7906, -1.9234, -6.6092,
       -7.3757, -0.1521,  1.4003, -0.8045,  0.376 , -1.8386,  0.7657,
        0.7104, -0.4229])
```

[47]:
```

```

5 clean the data

[48]:
```
# Clean the jobcity column by stripping whitespace and converting to lowercase
df['JobCity'] = df['JobCity'].str.strip().str.lower()

# Get the unique cleaned job cities
unique_cities_cleaned = df['JobCity'].unique()

# Print the unique cleaned cities
print(unique_cities_cleaned)
```

```
['bangalore' 'indore' 'chennai' 'gurgaon' 'manesar' 'hyderabad' 'banglore'
 'noida' 'kolkata' 'pune' '-1' 'mohali' 'jhansi' 'delhi' 'bhubaneswar'
 'navi mumbai' 'mumbai' 'new delhi' 'mangalore' 'rewari' 'gaziabaad'
 'bhiwadi' 'mysore' 'rajkot' 'greater noida' 'jaipur' 'thane'
 'maharajganj' 'thiruvananthapuram' 'punchkula' 'bhubaneshwar'
 'coimbatore' 'dhanbad' 'lucknow' 'trivandrum' 'gandhi nagar' 'una'
 'daman and diu' 'gurgoan' 'vsakhapttnam' 'nagpur' 'bhagalpur'
 'new delhi - jaisalmer' 'ahmedabad' 'kochi/cochin' 'bankura' 'bengaluru'
 'kanpur' 'vijayawada' 'kochi' 'beawar' 'alwar' 'siliguri' 'raipur'
 'bhopal' 'faridabad' 'jodhpur' 'udaipur' 'muzaffarpur' 'kolkata`'
 'bulandshahar' 'haridwar' 'raigarh' 'visakhapatnam' 'jabalpur' 'unnao'
 'aurangabad' 'belgaum' 'dehradun' 'rudrapur' 'jamshedpur' 'vizag' 'nouda'
 'dharamshala' 'banagalore' 'hissar' 'ranchi' 'madurai' 'gurga'
 'chandigarh' 'australia' 'cheyyar' 'sonepat' 'ghaziabad' 'pantnagar'
 'jagdalpur' 'angul' 'baroda' 'ariyalur' 'jowai'
 'kochi/cochin, chennai and coimbatore' 'neemrana' 'tirupathi'
 'bhubneshwar' 'calicut' 'gandhinagar' 'dubai' 'ahmednagar' 'nashik'
 'bellary' 'ludhiana' 'muzaffarnagar' 'gagret' 'indirapuram, ghaziabad'
 'gwalior' 'chennai & mumbai' 'rajasthan' 'sonipat' 'bareli' 'hospete'
 'miryalaguda' 'dharuhera' 'meerut' 'ganjam' 'hubli' 'ncr' 'agra' 'trichy'
 'kudankulam ,tarapur' 'ongole' 'sambalpur' 'pondicherry' 'bundi'
 'sadulpur,rajgarh,distt-churu,rajasthan' 'am' 'bikaner' 'vadodara'
 'india' 'asansol' 'tirunelvelli' 'ernakulam' 'bilaspur' 'chandrapur'
 'nanded' 'dharmapuri' 'vandavasi' 'rohtak' 'patna' 'salem' 'nasikcity'
 'technopark, trivandrum' 'bharuch' 'tornagallu' 'jaspur' 'burdwan'
 'shimla' 'gajiabaad' 'jammu' 'shahdol' 'muvattupuzha'
 'al jubail,saudi arabia' 'kalmar, sweden' 'secunderabad'
 'a-64,sec-64,noida' 'ratnagiri' 'jhajjar' 'gulbarga'
 'hyderabad(bhadurpally)' 'nalagarh' 'jeddah saudi arabia'
 'chennai, bangalore' 'jamnagar' 'tirupati' 'gonda' 'orissa' 'kharagpur'
 'navi mumbai , hyderabad' 'joshimath' 'bathinda' 'johannesburg'
 'kala amb' 'karnal' 'london' 'kota' 'panchkula' 'baddi hp' 'nagari'
 'mettur, tamil nadu' 'durgapur' 'pondi' 'surat' 'kurnool' 'kolhapur'
 'bhilai' 'hderabad' 'bahadurgarh' 'rayagada, odisha' 'kakinada'
 'varanasi' 'punr' 'nellore' 'sahibabad' 'howrah' 'trichur' 'ambala'
 'khopoli' 'keral' 'roorkee' 'allahabad' 'delhi/ncr' 'jalandhar' 'vapi'
 'pilani' 'muzzafarpur' 'ras al khaimah' 'bihar' 'singaruli' 'pondy'
```

```
'phagwara' 'guragaon' 'baripada' 'yamuna nagar' 'shahibabad' 'sampla'
'guwahati' 'rourkela' 'banaglore' 'vellore' 'dausa'
'latur (maharashtra )' 'mainpuri' 'dammam' 'haldia' 'rae bareli'
'patiala' 'gorakhpur' 'new dehli' 'ambala city' 'karad' 'rajpura'
'haryana' 'asifabadbanglore']
```

[49]:
```
# City mapping dictionary (all keys should be in lowercase)
city_mapping = {
    'bangalore': 'Bangalore',
    'banglore': 'Bangalore',
    'banagalore': 'Bangalore',
    'bengaluru': 'Bangalore',
    'asifabadbanglore': 'Bangalore',
    'indore': 'Indore',
    'chennai': 'Chennai',
    'gurgaon': 'Gurgaon',
    'gurgoan': 'Gurgaon',
    'gurga': 'Gurgaon',
    'manesar': 'Manesar',
    'hyderabad': 'Hyderabad',
    'hderabad': 'Hyderabad',
    'hyderabad(bhadurpally)': 'Hyderabad',
    'noida': 'Noida',
    'nouda': 'Noida',
    'kolkata': 'Kolkata',
    'kolkata`': 'Kolkata',
    'pune': 'Pune',
    '-1': 'Unknown',
    'mohali': 'Mohali',
    'jhansi': 'Jhansi',
    'delhi': 'Delhi',
    'new delhi': 'New Delhi',
    'bhubaneswar': 'Bhubaneswar',
    'bhubaneshwar': 'Bhubaneswar',
    'navi mumbai': 'Navi Mumbai',
    'mumbai': 'Mumbai',
    'mangalore': 'Mangalore',
    'rewari': 'Rewari',
    'gaziabaad': 'Ghaziabad',
    'ghaziabad': 'Ghaziabad',
    'bhiwadi': 'Bhiwadi',
    'mysore': 'Mysore',
    'rajkot': 'Rajkot',
    'greater noida': 'Greater Noida',
    'jaipur': 'Jaipur',
    'thane': 'Thane',
    'maharajganj': 'Maharajganj',
```

```
    'thiruvananthapuram': 'Thiruvananthapuram',
    'punchkula': 'Panchkula',
    'coimbatore': 'Coimbatore',
    'dhanbad': 'Dhanbad',
    'lucknow': 'Lucknow',
    'trivandrum': 'Thiruvananthapuram',
    'gandhi nagar': 'Gandhinagar',
    'una': 'Una',
    'daman and diu': 'Daman and Diu',
    'vsakhapttnam': 'Visakhapatnam',
    'nagpur': 'Nagpur',
    'bhagalpur': 'Bhagalpur',
    'new delhi- jaisalmer': 'New Delhi',
    'ahmedabad': 'Ahmedabad',
    'kochi/cochin': 'Kochi',
    'bankura': 'Bankura',
    'kanpur': 'Kanpur',
    'vijayawada': 'Vijayawada',
    'kochi': 'Kochi',
    'beawar': 'Beawar',
    'alwar': 'Alwar',
    'siliguri': 'Siliguri',
    'raipur': 'Raipur',
    'bhopal': 'Bhopal',
    'faridabad': 'Faridabad',
    'jodhpur': 'Jodhpur',
    'udaipur': 'Udaipur',
    'muzaffarpur': 'Muzaffarpur',
    'bulandshahar': 'Bulandshahar',
    'haridwar': 'Haridwar',
    'raigarh': 'Raigarh',
    'visakhapatnam': 'Visakhapatnam',
    'jabalpur': 'Jabalpur',
    'unnao': 'Unnao',
    'aurangabad': 'Aurangabad',
    'belgaum': 'Belgaum',
    'dehradun': 'Dehradun',
    'rudrapur': 'Rudrapur',
    'jamshedpur': 'Jamshedpur',
    'vizag': 'Visakhapatnam',
    'dharamshala': 'Dharamshala',
    'hissar': 'Hisar',
    'ranchi': 'Ranchi',
    'madurai': 'Madurai',
    'chandigarh': 'Chandigarh',
    'australia': 'Australia',
    'cheyyar': 'Cheyyar',
```

```
    'sonepat': 'Sonepat',
    'pantnagar': 'Pantnagar',
    'jagdalpur': 'Jagdalpur',
    'angul': 'Angul',
    'baroda': 'Vadodara',
    'ariyalur': 'Ariyalur',
    'jowai': 'Jowai',
    'neemrana': 'Neemrana',
    'tirupathi': 'Tirupati',
    'bhubneshwar': 'Bhubaneswar',
    'calicut': 'Kozhikode',
    'gandhinagar': 'Gandhinagar',
    'dubai': 'Dubai',
    'ahmednagar': 'Ahmednagar',
    'nashik': 'Nashik',
    'bellary': 'Bellary',
    'ludhiana': 'Ludhiana',
    'muzaffarnagar': 'Muzaffarnagar',
    'gagret': 'Gagret',
    'indirapuram, ghaziabad': 'Ghaziabad',
    'gwalior': 'Gwalior',
    'chennai & mumbai': 'Chennai',
    'rajasthan': 'Rajasthan',
    'sonipat': 'Sonipat',
    'bareli': 'Bareli',
    'hospete': 'Hospete',
    'miryalaguda': 'Miryalaguda',
    'dharuhera': 'Dharuhera',
    'meerut': 'Meerut',
    'ganjam': 'Ganjam',
    'hubli': 'Hubli',
    'ncr': 'NCR',
    'agra': 'Agra',
    'trichy': 'Tiruchirappalli',
    'kudankulam ,tarapur': 'Kudankulam',
    'ongole': 'Ongole',
    'sambalpur': 'Sambalpur',
    'pondicherry': 'Puducherry',
    'bundi': 'Bundi',
    'sadulpur,rajgarh,distt-churu,rajasthan': 'Rajasthan',
    'am': 'Am',
    'bikaner': 'Bikaner',
    'vadodara': 'Vadodara',
    'india': 'India',
    'asansol': 'Asansol',
    'tirunelvelli': 'Tirunelvelli',
    'ernakulam': 'Ernakulam',
```

```
    'bilaspur': 'Bilaspur',
    'chandrapur': 'Chandrapur',
    'nanded': 'Nanded',
    'dharmapuri': 'Dharmapuri',
    'vandavasi': 'Vandavasi',
    'rohtak': 'Rohtak',
    'patna': 'Patna',
    'salem': 'Salem',
    'nasikcity': 'Nashik',
    'technopark, trivandrum': 'Trivandrum',
    'bharuch': 'Bharuch',
    'tornagallu': 'Tornagallu',
    'jaspur': 'Jaspur',
    'burdwan': 'Burdwan',
    'shimla': 'Shimla',
    'gajiabaad': 'Ghaziabad',
    'jammu': 'Jammu',
    'shahdol': 'Shahdol',
    'muvattupuzha': 'Muvattupuzha',
    'al jubail,saudi arabia': 'Al Jubail',
    'kalmar, sweden': 'Kalmar',
    'secunderabad': 'Secunderabad',
    'a-64,sec-64,noida': 'Noida',
    'ratnagiri': 'Ratnagiri',
    'jhajjar': 'Jhajjar',
    'gulbarga': 'Gulbarga',
    'hyderabad(bhadurpally)': 'Hyderabad',
    'nalagarh': 'Nalagarh',
    'jeddah saudi arabia': 'Jeddah',
    'chennai, bangalore': 'Chennai',
    'jamnagar': 'Jamnagar',
    'tirupati': 'Tirupati',
    'gonda': 'Gonda',
    'orissa': 'Odisha',
    'kharagpur': 'Kharagpur',
    'navi mumbai , hyderabad': 'Navi Mumbai',
    'joshimath': 'Joshimath',
    'bathinda': 'Bathinda',
    'johannesburg': 'Johannesburg',
    'kala amb': 'Kala Amb',
    'karnal': 'Karnal',
    'london': 'London',
    'kota': 'Kota',
    'dehraj': 'Dehradun',
    'melbourne': 'Melbourne',
    'moradabad': 'Moradabad',
    'delhi-gurgaon': 'Delhi',
```

```
    'ambala': 'Ambala',
    'faridkot': 'Faridkot',
    'rohtak, haryana': 'Rohtak',
    'khammam': 'Khammam',
    'khurda': 'Khurda',
    'jhalawar': 'Jhalawar',
    'kaithal': 'Kaithal',
    'sonbhadra': 'Sonbhadra',
    'fatehgarh sahib': 'Fatehgarh Sahib',
    'kaithal-haryana': 'Kaithal',
    'bhilwara': 'Bhilwara',
    'coimbatore, tirupur': 'Coimbatore',
    'sri ganganagar': 'Sri Ganganagar',
    'manipal': 'Manipal',
    'tirupathi': 'Tirupati',
    'kharagpur, west bengal': 'Kharagpur',
    'kolkata': 'Kolkata',
    'trichy-tiruchirappalli': 'Tiruchirappalli',
}

# Convert jobcity values to lowercase


# Replace jobcity values using the city_mapping dictionary
df['JobCity'] = df['JobCity'].replace(city_mapping)
df['JobCity'] = df['JobCity'].str.strip().str.lower()
# Check the updated jobcity values
print(df['JobCity'].unique())
```

```
['bangalore' 'indore' 'chennai' 'gurgaon' 'manesar' 'hyderabad' 'noida'
 'kolkata' 'pune' 'unknown' 'mohali' 'jhansi' 'delhi' 'bhubaneswar'
 'navi mumbai' 'mumbai' 'new delhi' 'mangalore' 'rewari' 'ghaziabad'
 'bhiwadi' 'mysore' 'rajkot' 'greater noida' 'jaipur' 'thane'
 'maharajganj' 'thiruvananthapuram' 'panchkula' 'coimbatore' 'dhanbad'
 'lucknow' 'gandhinagar' 'una' 'daman and diu' 'visakhapatnam' 'nagpur'
 'bhagalpur' 'new delhi - jaisalmer' 'ahmedabad' 'kochi' 'bankura'
 'kanpur' 'vijayawada' 'beawar' 'alwar' 'siliguri' 'raipur' 'bhopal'
 'faridabad' 'jodhpur' 'udaipur' 'muzaffarpur' 'bulandshahar' 'haridwar'
 'raigarh' 'jabalpur' 'unnao' 'aurangabad' 'belgaum' 'dehradun' 'rudrapur'
 'jamshedpur' 'dharamshala' 'hisar' 'ranchi' 'madurai' 'chandigarh'
 'australia' 'cheyyar' 'sonepat' 'pantnagar' 'jagdalpur' 'angul'
 'vadodara' 'ariyalur' 'jowai' 'kochi/cochin, chennai and coimbatore'
 'neemrana' 'tirupati' 'kozhikode' 'dubai' 'ahmednagar' 'nashik' 'bellary'
 'ludhiana' 'muzaffarnagar' 'gagret' 'gwalior' 'rajasthan' 'sonipat'
 'bareli' 'hospete' 'miryalaguda' 'dharuhera' 'meerut' 'ganjam' 'hubli'
 'ncr' 'agra' 'tiruchirappalli' 'kudankulam' 'ongole' 'sambalpur'
 'puducherry' 'bundi' 'am' 'bikaner' 'india' 'asansol' 'tirunelvelli'
```

```
'ernakulam' 'bilaspur' 'chandrapur' 'nanded' 'dharmapuri' 'vandavasi'
'rohtak' 'patna' 'salem' 'trivandrum' 'bharuch' 'tornagallu' 'jaspur'
'burdwan' 'shimla' 'jammu' 'shahdol' 'muvattupuzha' 'al jubail' 'kalmar'
'secunderabad' 'ratnagiri' 'jhajjar' 'gulbarga' 'nalagarh' 'jeddah'
'jamnagar' 'gonda' 'odisha' 'kharagpur' 'joshimath' 'bathinda'
'johannesburg' 'kala amb' 'karnal' 'london' 'kota' 'baddi hp' 'nagari'
'mettur, tamil nadu' 'durgapur' 'pondi' 'surat' 'kurnool' 'kolhapur'
'bhilai' 'bahadurgarh' 'rayagada, odisha' 'kakinada' 'varanasi' 'punr'
'nellore' 'sahibabad' 'howrah' 'trichur' 'ambala' 'khopoli' 'keral'
'roorkee' 'allahabad' 'delhi/ncr' 'jalandhar' 'vapi' 'pilani'
'muzzafarpur' 'ras al khaimah' 'bihar' 'singaruli' 'pondy' 'phagwara'
'guragaon' 'baripada' 'yamuna nagar' 'shahibabad' 'sampla' 'guwahati'
'rourkela' 'banaglore' 'vellore' 'dausa' 'latur (maharashtra )'
'mainpuri' 'dammam' 'haldia' 'rae bareli' 'patiala' 'gorakhpur'
'new dehli' 'ambala city' 'karad' 'rajpura' 'haryana']
```

[50]:
```
df
```

[50]:

	ID	Salary	DOJ	DOL	\
0	203097	420000.0	6/1/12 0:00	present	
1	579905	500000.0	9/1/13 0:00	present	
2	810601	325000.0	6/1/14 0:00	present	
3	267447	1100000.0	7/1/11 0:00	present	
4	343523	200000.0	3/1/14 0:00	3/1/15 0:00	
…	…	…	…	…	
3993	47916	280000.0	10/1/11 0:00	10/1/12 0:00	
3994	752781	100000.0	7/1/13 0:00	7/1/13 0:00	
3995	355888	320000.0	7/1/13 0:00	present	
3996	947111	200000.0	7/1/14 0:00	1/1/15 0:00	
3997	324966	400000.0	2/1/13 0:00	present	

	Designation	JobCity	Gender	DOB	\
0	senior quality engineer	bangalore	f	2/19/90 0:00	
1	assistant manager	indore	m	10/4/89 0:00	
2	systems engineer	chennai	f	8/3/92 0:00	
3	senior software engineer	gurgaon	m	12/5/89 0:00	
4	get	manesar	m	2/27/91 0:00	
…	…	…	…	…	
3993	software engineer	new delhi	m	4/15/87 0:00	
3994	technical writer	hyderabad	f	8/27/92 0:00	
3995	associate software engineer	bangalore	m	7/3/91 0:00	
3996	software developer	bangalore	f	3/20/92 0:00	
3997	senior systems engineer	chennai	f	2/26/91 0:00	

	10percentage	10board	…	ComputerScience	\
0	84.30	board ofsecondary education,ap	…	-1	
1	85.40	cbse	…	-1	

```
2           85.00                          cbse  …              -1
3           85.60                          cbse  …              -1
4           78.00                          cbse  …              -1
…             …                             …  …             …
3993        52.09                          cbse  …              -1
3994        90.00                   state board  …              -1
3995        81.86                    bse,odisha  …              -1
3996        78.72                   state board  …             438
3997        70.60                          cbse  …              -1

      MechanicalEngg ElectricalEngg  TelecomEngg  CivilEngg conscientiousness  \
0                 -1             -1           -1         -1            0.9737
1                 -1             -1           -1         -1           -0.7335
2                 -1             -1           -1         -1            0.2718
3                 -1             -1           -1         -1            0.0464
4                 -1             -1           -1         -1           -0.8810
…               …              …            …          …              …
3993              -1             -1           -1         -1           -0.1082
3994              -1             -1           -1         -1           -0.3027
3995              -1             -1           -1         -1           -1.5765
3996              -1             -1           -1         -1           -0.1590
3997              -1             -1           -1         -1           -1.1128

     agreeableness  extraversion  nueroticism  openess_to_experience
0           0.8128        0.5269      1.35490                -0.4455
1           0.3789        1.2396     -0.10760                 0.8637
2           1.7109        0.1637     -0.86820                 0.6721
3           0.3448       -0.3440     -0.40780                -0.9194
4          -0.2793       -1.0697      0.09163                -0.1295
…            …            …           …                     …
3993        0.3448        0.2366      0.64980                -0.9194
3994        0.8784        0.9322      0.77980                -0.0943
3995       -1.5273       -1.5051     -1.31840                -0.7615
3996        0.0459       -0.4511     -0.36120                -0.0943
3997       -0.2793       -0.6343      1.32553                -0.6035

[3998 rows x 38 columns]
```

[51]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 38 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     3998 non-null   int64
 1   Salary                 3998 non-null   float64
```

```
 2   DOJ                    3998 non-null   object
 3   DOL                    3998 non-null   object
 4   Designation            3998 non-null   object
 5   JobCity                3998 non-null   object
 6   Gender                 3998 non-null   object
 7   DOB                    3998 non-null   object
 8   10percentage           3998 non-null   float64
 9   10board                3998 non-null   object
 10  12graduation           3998 non-null   int64
 11  12percentage           3998 non-null   float64
 12  12board                3998 non-null   object
 13  CollegeID              3998 non-null   int64
 14  CollegeTier            3998 non-null   int64
 15  Degree                 3998 non-null   object
 16  Specialization         3998 non-null   object
 17  collegeGPA             3998 non-null   float64
 18  CollegeCityID          3998 non-null   int64
 19  CollegeCityTier        3998 non-null   int64
 20  CollegeState           3998 non-null   object
 21  GraduationYear         3998 non-null   int64
 22  English                3998 non-null   int64
 23  Logical                3998 non-null   int64
 24  Quant                  3998 non-null   int64
 25  Domain                 3998 non-null   float64
 26  ComputerProgramming    3998 non-null   int64
 27  ElectronicsAndSemicon  3998 non-null   int64
 28  ComputerScience        3998 non-null   int64
 29  MechanicalEngg         3998 non-null   int64
 30  ElectricalEngg         3998 non-null   int64
 31  TelecomEngg            3998 non-null   int64
 32  CivilEngg              3998 non-null   int64
 33  conscientiousness      3998 non-null   float64
 34  agreeableness          3998 non-null   float64
 35  extraversion           3998 non-null   float64
 36  nueroticism            3998 non-null   float64
 37  openess_to_experience  3998 non-null   float64
dtypes: float64(10), int64(17), object(11)
memory usage: 1.2+ MB
```

[52]:
```
df['DOJ'] = pd.to_datetime(df['DOJ'], errors='coerce')  # Use errors='coerce'␣
↪to handle invalid dates
```

```
<ipython-input-52-e279bed6f173>:1: UserWarning: Could not infer format, so each
element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df['DOJ'] = pd.to_datetime(df['DOJ'], errors='coerce')  # Use errors='coerce'
to handle invalid dates
```

[53]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 38 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     3998 non-null   int64
 1   Salary                 3998 non-null   float64
 2   DOJ                    3998 non-null   datetime64[ns]
 3   DOL                    3998 non-null   object
 4   Designation            3998 non-null   object
 5   JobCity                3998 non-null   object
 6   Gender                 3998 non-null   object
 7   DOB                    3998 non-null   object
 8   10percentage           3998 non-null   float64
 9   10board                3998 non-null   object
 10  12graduation           3998 non-null   int64
 11  12percentage           3998 non-null   float64
 12  12board                3998 non-null   object
 13  CollegeID              3998 non-null   int64
 14  CollegeTier            3998 non-null   int64
 15  Degree                 3998 non-null   object
 16  Specialization         3998 non-null   object
 17  collegeGPA             3998 non-null   float64
 18  CollegeCityID          3998 non-null   int64
 19  CollegeCityTier        3998 non-null   int64
 20  CollegeState           3998 non-null   object
 21  GraduationYear         3998 non-null   int64
 22  English                3998 non-null   int64
 23  Logical                3998 non-null   int64
 24  Quant                  3998 non-null   int64
 25  Domain                 3998 non-null   float64
 26  ComputerProgramming    3998 non-null   int64
 27  ElectronicsAndSemicon  3998 non-null   int64
 28  ComputerScience        3998 non-null   int64
 29  MechanicalEngg         3998 non-null   int64
 30  ElectricalEngg         3998 non-null   int64
 31  TelecomEngg            3998 non-null   int64
 32  CivilEngg              3998 non-null   int64
 33  conscientiousness      3998 non-null   float64
 34  agreeableness          3998 non-null   float64
 35  extraversion           3998 non-null   float64
 36  nueroticism            3998 non-null   float64
 37  openess_to_experience  3998 non-null   float64
dtypes: datetime64[ns](1), float64(10), int64(17), object(10)
memory usage: 1.2+ MB
```

[54]:
```
df.shape
```

[54]:
```
(3998, 38)
```

[55]:
```
df['DOL'].head()
```

[55]:
```
0        present
1        present
2        present
3        present
4    3/1/15 0:00
Name: DOL, dtype: object
```

[56]:
```
df['DOL'] = df['DOL'].apply(lambda x: "Left" if x != "present" else x)
```

[57]:
```
df['DOL'].head()
```

[57]:
```
0    present
1    present
2    present
3    present
4       Left
Name: DOL, dtype: object
```

[58]:
```
df.head()
```

[58]:

	ID	Salary	DOJ	DOL	Designation	JobCity	\
0	203097	420000.0	2012-06-01	present	senior quality engineer	bangalore	
1	579905	500000.0	2013-09-01	present	assistant manager	indore	
2	810601	325000.0	2014-06-01	present	systems engineer	chennai	
3	267447	1100000.0	2011-07-01	present	senior software engineer	gurgaon	
4	343523	200000.0	2014-03-01	Left	get	manesar	

	Gender	DOB	10percentage	10board	…	\
0	f	2/19/90 0:00	84.3	board ofsecondary education,ap	…	
1	m	10/4/89 0:00	85.4	cbse	…	
2	f	8/3/92 0:00	85.0	cbse	…	
3	m	12/5/89 0:00	85.6	cbse	…	
4	m	2/27/91 0:00	78.0	cbse	…	

	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 38 columns]
```

[59]:
```
df['DOL'].value_counts()
```

[59]:
```
DOL
Left       2123
present    1875
Name: count, dtype: int64
```

[60]:
```
df['Salary'].describe()
```

[60]:
```
count    3.998000e+03
mean     3.076998e+05
std      2.127375e+05
min      3.500000e+04
25%      1.800000e+05
50%      3.000000e+05
75%      3.700000e+05
max      4.000000e+06
Name: Salary, dtype: float64
```

[61]:
```
# List of columns to replace -1 with 0
columns_to_replace = [
    'ComputerScience',
    'MechanicalEngg',
    'ElectricalEngg',
    'TelecomEngg',
    'CivilEngg'
]

# Replace -1 with 0 for each specified column
for column in columns_to_replace:
    df[column] = df[column].replace(-1, 0)
```

[62]:
```
# Replace -1 with 0 in specified columns
df[columns_to_replace] = df[columns_to_replace].replace(-1, 0)
```

[63]:
```
df.head()
```

[63]:

	ID	Salary	DOJ	DOL	Designation	JobCity	\
0	203097	420000.0	2012-06-01	present	senior quality engineer	bangalore	
1	579905	500000.0	2013-09-01	present	assistant manager	indore	
2	810601	325000.0	2014-06-01	present	systems engineer	chennai	
3	267447	1100000.0	2011-07-01	present	senior software engineer	gurgaon	
4	343523	200000.0	2014-03-01	Left	get	manesar	

	Gender	DOB	10percentage	10board	…	\
0	f	2/19/90 0:00	84.3	board ofsecondary education,ap	…	
1	m	10/4/89 0:00	85.4	cbse	…	
2	f	8/3/92 0:00	85.0	cbse	…	
3	m	12/5/89 0:00	85.6	cbse	…	
4	m	2/27/91 0:00	78.0	cbse	…	

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

	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 38 columns]

[64]:
```
df.describe().transpose()
```

[64]:

	count	mean	\
ID	3998.0	663794.54052	
Salary	3998.0	307699.849925	

DOJ	3998	2013-07-02 11:04:10.325162496
10percentage	3998.0	77.925443
12graduation	3998.0	2008.087544
12percentage	3998.0	74.466366
CollegeID	3998.0	5156.851426
CollegeTier	3998.0	1.925713
collegeGPA	3998.0	71.486171
CollegeCityID	3998.0	5156.851426
CollegeCityTier	3998.0	0.3004
GraduationYear	3998.0	2012.105803
English	3998.0	501.649075
Logical	3998.0	501.598799
Quant	3998.0	513.378189
Domain	3998.0	0.51049
ComputerProgramming	3998.0	353.102801
ElectronicsAndSemicon	3998.0	95.328414
ComputerScience	3998.0	91.516758
MechanicalEngg	3998.0	23.915958
ElectricalEngg	3998.0	17.438469
TelecomEngg	3998.0	32.757629
CivilEngg	3998.0	3.673337
conscientiousness	3998.0	-0.037831
agreeableness	3998.0	0.146496
extraversion	3998.0	0.002763
nueroticism	3998.0	-0.169033
openess_to_experience	3998.0	-0.13811

	min	25%	\
ID	11244.0	334284.25	
Salary	35000.0	180000.0	
DOJ	1991-06-01 00:00:00	2012-10-01 00:00:00	
10percentage	43.0	71.68	
12graduation	1995.0	2007.0	
12percentage	40.0	66.0	
CollegeID	2.0	494.0	
CollegeTier	1.0	2.0	
collegeGPA	6.45	66.4075	
CollegeCityID	2.0	494.0	
CollegeCityTier	0.0	0.0	
GraduationYear	0.0	2012.0	
English	180.0	425.0	
Logical	195.0	445.0	
Quant	120.0	430.0	
Domain	-1.0	0.342315	
ComputerProgramming	-1.0	295.0	
ElectronicsAndSemicon	-1.0	-1.0	
ComputerScience	0.0	0.0	

			\
MechanicalEngg	0.0	0.0	
ElectricalEngg	0.0	0.0	
TelecomEngg	0.0	0.0	
CivilEngg	0.0	0.0	
conscientiousness	-4.1267	-0.713525	
agreeableness	-5.7816	-0.2871	
extraversion	-4.6009	-0.6048	
nueroticism	-2.643	-0.8682	
openess_to_experience	-7.3757	-0.6692	

	50%	75%	\
ID	639600.0	990480.0	
Salary	300000.0	370000.0	
DOJ	2013-11-01 00:00:00	2014-07-01 00:00:00	
10percentage	79.15	85.67	
12graduation	2008.0	2009.0	
12percentage	74.4	82.6	
CollegeID	3879.0	8818.0	
CollegeTier	2.0	2.0	
collegeGPA	71.72	76.3275	
CollegeCityID	3879.0	8818.0	
CollegeCityTier	0.0	1.0	
GraduationYear	2013.0	2014.0	
English	500.0	570.0	
Logical	505.0	565.0	
Quant	515.0	595.0	
Domain	0.622643	0.842248	
ComputerProgramming	415.0	495.0	
ElectronicsAndSemicon	-1.0	233.0	
ComputerScience	0.0	0.0	
MechanicalEngg	0.0	0.0	
ElectricalEngg	0.0	0.0	
TelecomEngg	0.0	0.0	
CivilEngg	0.0	0.0	
conscientiousness	0.0464	0.7027	
agreeableness	0.2124	0.8128	
extraversion	0.0914	0.672	
nueroticism	-0.2344	0.5262	
openess_to_experience	-0.0943	0.5024	

	max	std
ID	1298275.0	363218.245829
Salary	4000000.0	212737.499957
DOJ	2015-12-01 00:00:00	NaN
10percentage	97.76	9.850162
12graduation	2013.0	1.653599
12percentage	98.7	10.999933

CollegeID	18409.0	4802.261482
CollegeTier	2.0	0.26227
collegeGPA	99.93	8.167338
CollegeCityID	18409.0	4802.261482
CollegeCityTier	1.0	0.458489
GraduationYear	2017.0	31.857271
English	875.0	104.940021
Logical	795.0	86.783297
Quant	900.0	122.302332
Domain	0.99991	0.468671
ComputerProgramming	840.0	205.355519
ElectronicsAndSemicon	612.0	158.241218
ComputerScience	715.0	174.867677
MechanicalEngg	623.0	97.893295
ElectricalEngg	676.0	87.394072
TelecomEngg	548.0	104.568796
CivilEngg	516.0	36.559052
conscientiousness	1.9953	1.028666
agreeableness	1.9048	0.941782
extraversion	2.5354	0.951471
nueroticism	3.3525	1.00758
openess_to_experience	1.8224	1.008075

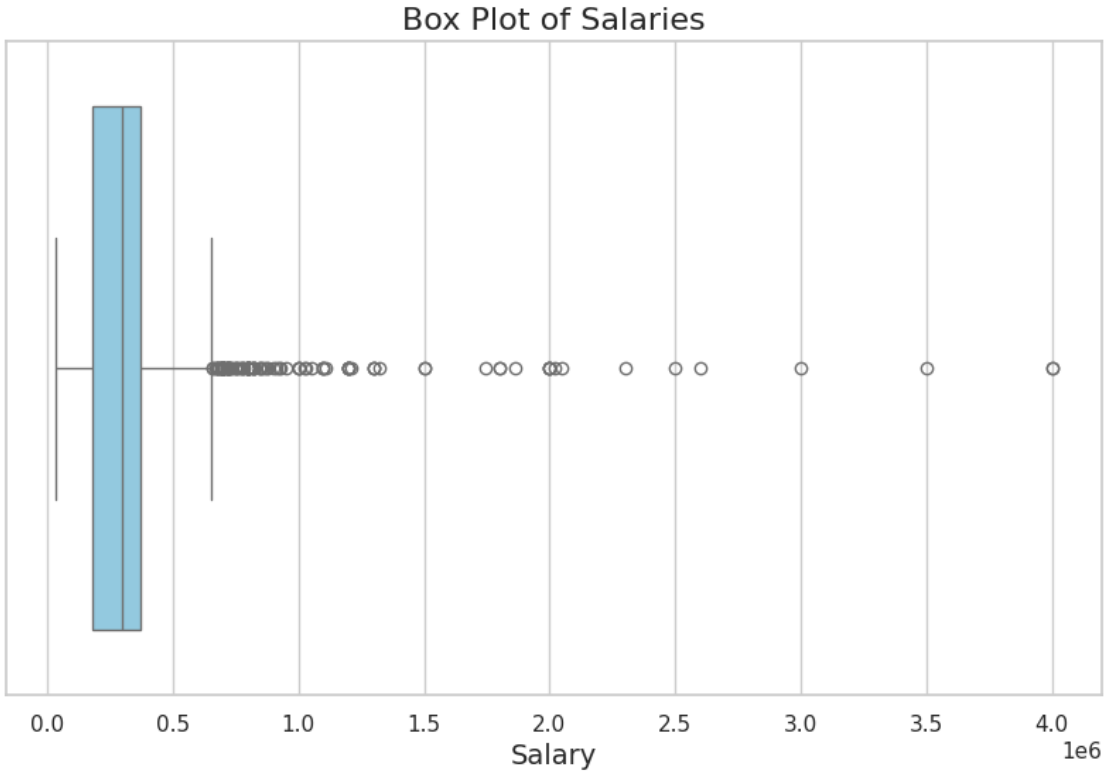
[64]:

6 step-3

[65]:
```
# Make sure 'Salary' is in a numeric format; remove commas and convert to float
df['Salary'] = df['Salary'].replace(',', '', regex=True).astype(float)

# Set the style for the plot
sns.set(style="whitegrid")

# Create a box plot for Salary
plt.figure(figsize=(10, 6))
sns.boxplot(x=df['Salary'], color='skyblue')

# Set the title and labels
plt.title('Box Plot of Salaries', fontsize=16)
plt.xlabel('Salary', fontsize=14)

# Show the plot
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:640:
FutureWarning: SeriesGroupBy.grouper is deprecated and will be removed in a
```

```
future version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```



7 INSIGHTS

Median Salary: The median salary, represented by the horizontal line inside the box, is around 0.4 million. This means that half of the individuals in the dataset earn less than 0.4 million and half earn more.

Interquartile Range (IQR): The IQR, represented by the box itself, is the range between the 25th and 75th percentiles. In this case, the IQR is relatively small, indicating that the middle 50% of salaries are clustered together.

Whiskers: The whiskers extend from the box to the minimum and maximum values excluding outliers. The length of the whiskers provides information about the spread of the data. In this case, the whiskers are relatively long, suggesting that there is a significant range of salaries in the dataset.

Outliers: The plot shows several outliers, which are data points that fall outside of the whiskers. These outliers represent individuals with exceptionally high or low salaries compared to the rest of the sample.

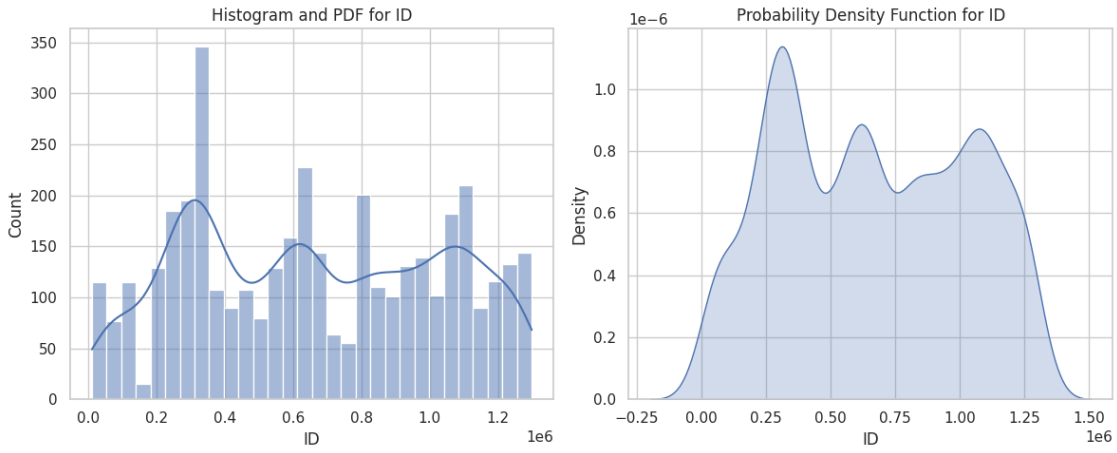
Overall Distribution: Based on the boxplot, the distribution of salaries is highly skewed to the right, with a long tail representing the high earners. The majority of salaries are concentrated in

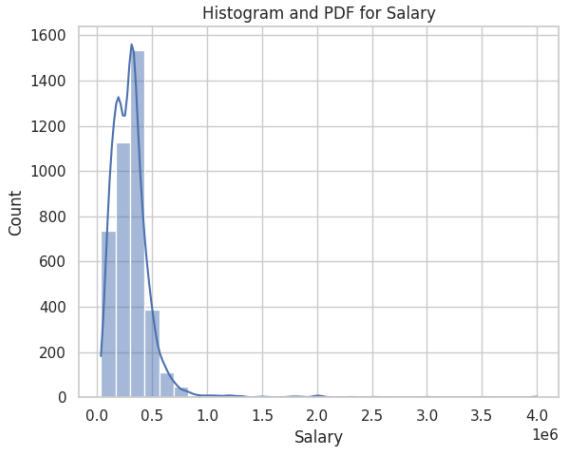
the lower range, with a smaller number of individuals earning significantly higher amounts.

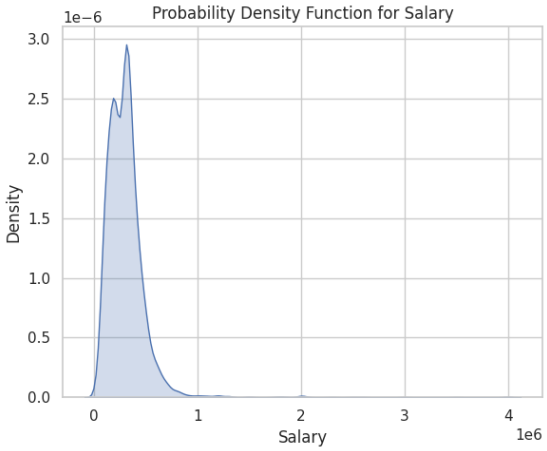
Additional Considerations:

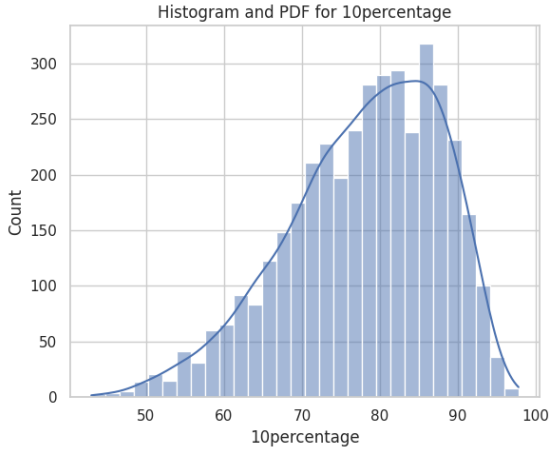
The specific units of the salary variable are not provided, so it is difficult to interpret the values in a meaningful context. The sample size is not known, which limits the ability to draw definitive conclusions about the population. In conclusion, the boxplot shows that the distribution of salaries is highly skewed to the right, with a small number of individuals earning significantly higher amounts than the majority. The median salary is around 0.4 million, and the interquartile range is relatively small, indicating that the middle 50% of salaries are clustered together. The presence of outliers suggests that there are some individuals with exceptionally high or low salaries.

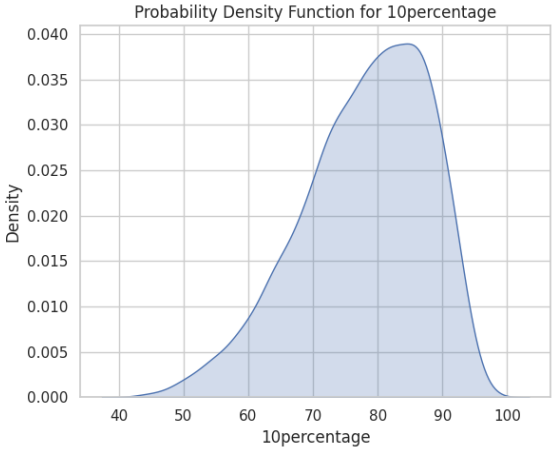
[66]:
```
# Assuming 'df' is your DataFrame
numerical_columns = df.select_dtypes(include=['number']).columns

# Create histograms and PDF plots for each numerical column
for col in numerical_columns:
    plt.figure(figsize=(12, 5))

    # Histogram
    plt.subplot(1, 2, 1)
    sns.histplot(df[col], bins=30, kde=True)
    plt.title(f'Histogram and PDF for {col}')

    # PDF
    plt.subplot(1, 2, 2)
    sns.kdeplot(df[col], fill=True)
    plt.title(f'Probability Density Function for {col}')

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

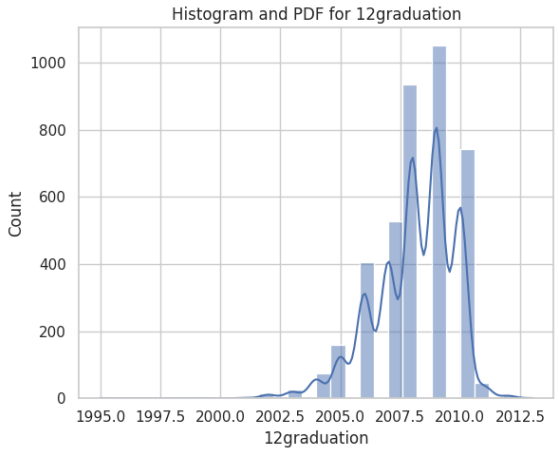


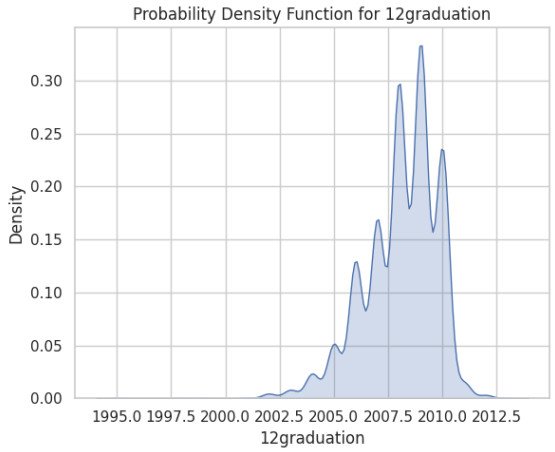

Histogram and PDF for Salary
Count
Salary
1e6

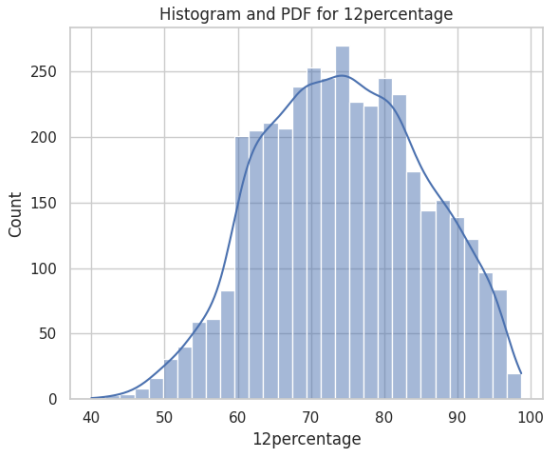

Probability Density Function for Salary
1e−6
Density
Salary
1e6

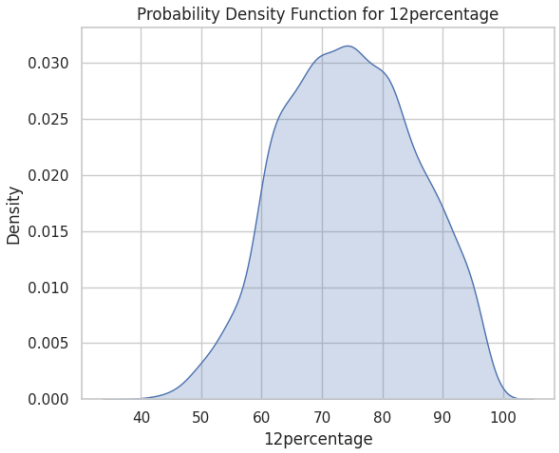

Histogram and PDF for 10percentage
Count
10percentage

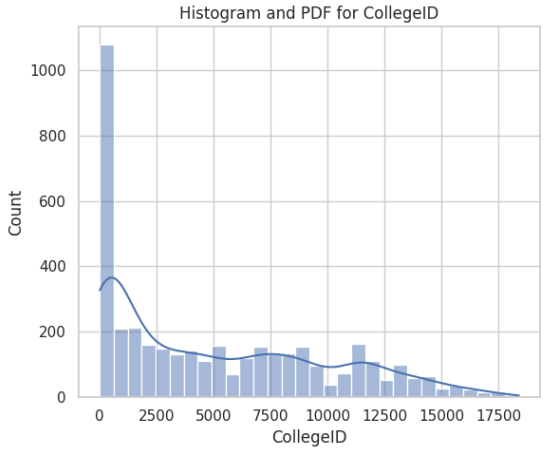

Probability Density Function for 10percentage
Density
10percentage

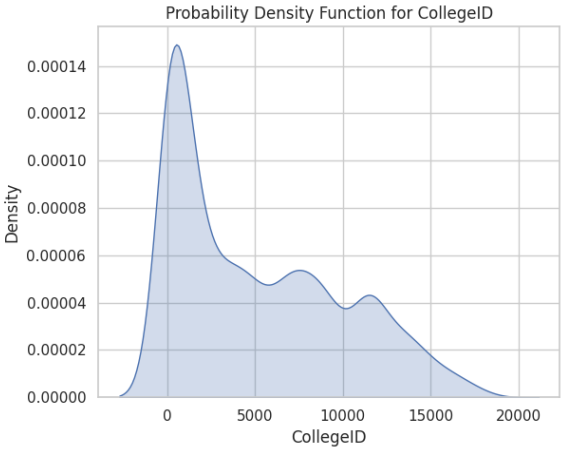

Histogram and PDF for 12graduation
Count
12graduation

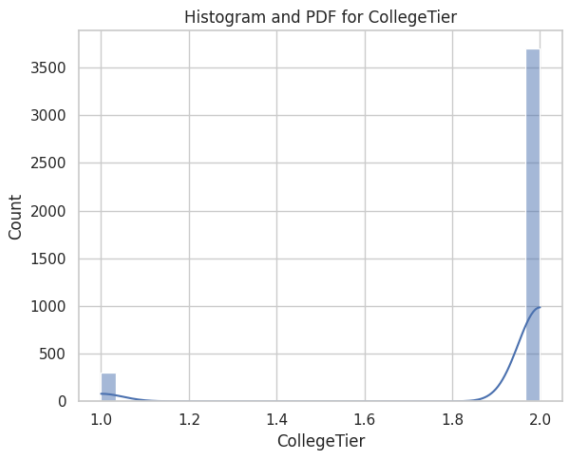

Probability Density Function for 12graduation
Density
12graduation

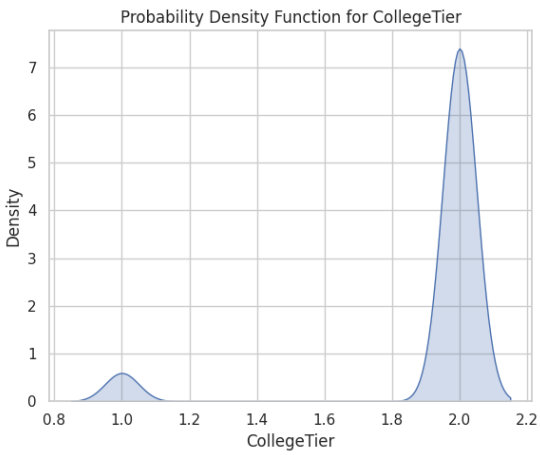

Histogram and PDF for 12percentage
Count
12percentage

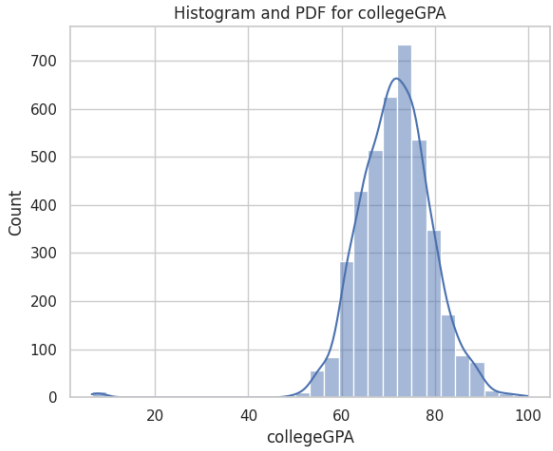

Probability Density Function for 12percentage
Density
12percentage

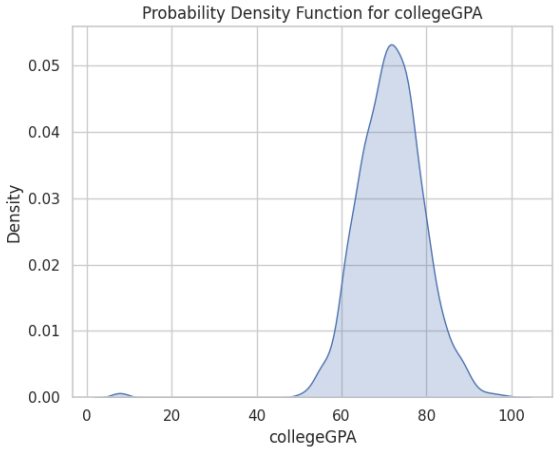

Histogram and PDF for CollegeID
Count
CollegeID

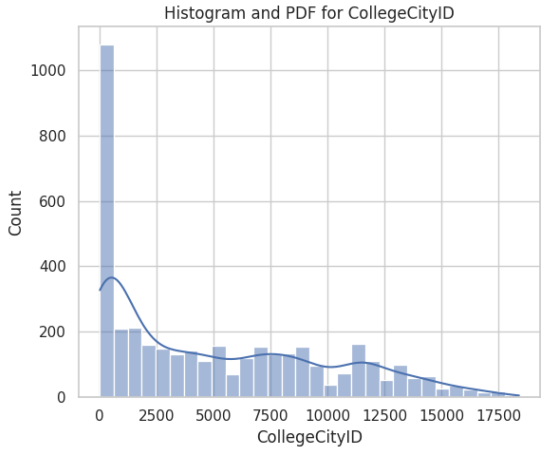

Probability Density Function for CollegeID
Density
CollegeID

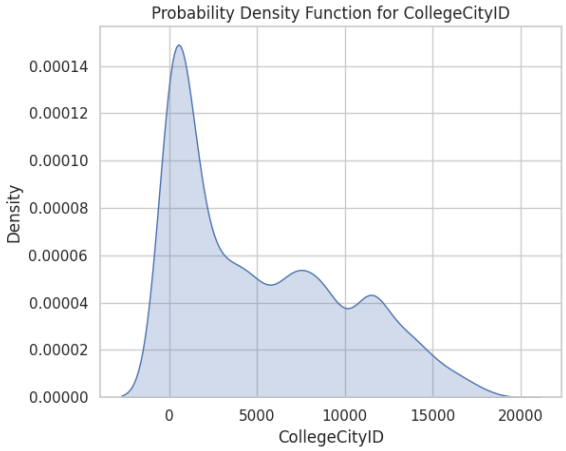

Histogram and PDF for CollegeTier
Count
CollegeTier

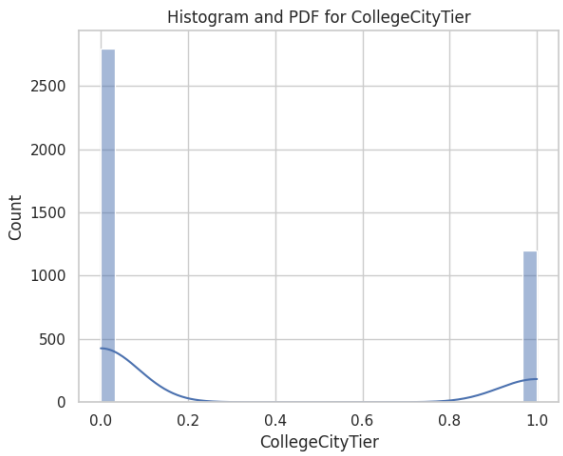

Probability Density Function for CollegeTier
Density
CollegeTier

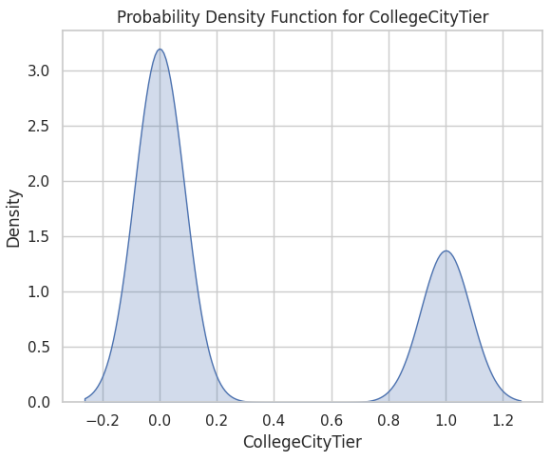

Histogram and PDF for collegeGPA
Count
collegeGPA

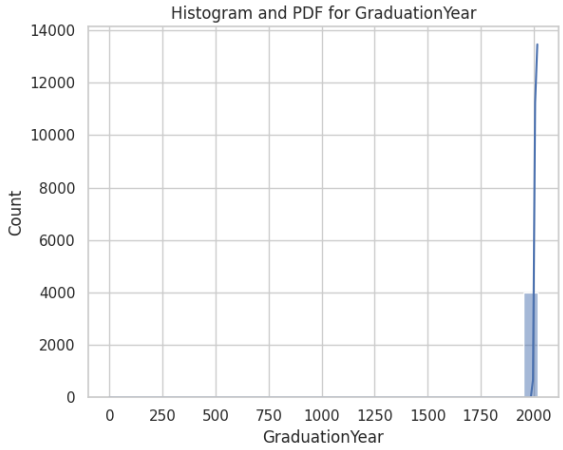

Probability Density Function for collegeGPA
Density
collegeGPA

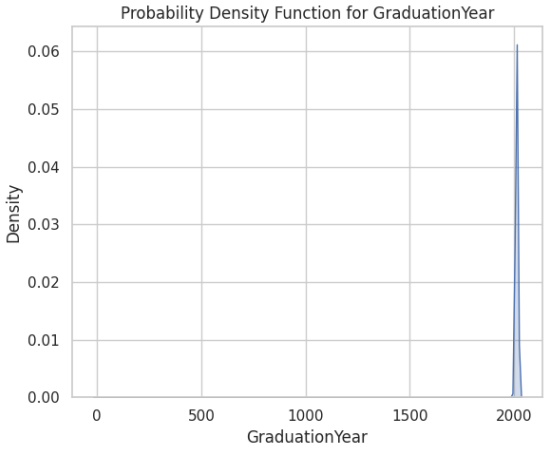

Histogram and PDF for CollegeCityID
Count
CollegeCityID

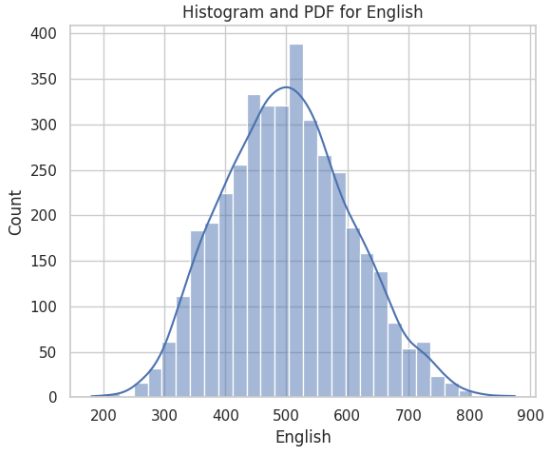

Probability Density Function for CollegeCityID
Density
CollegeCityID

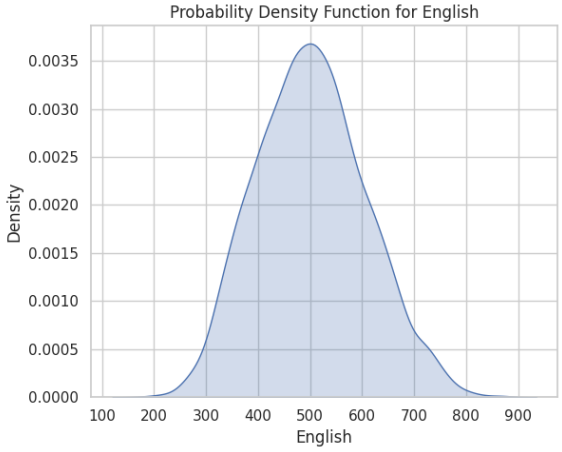

Histogram and PDF for CollegeCityTier
Count
CollegeCityTier

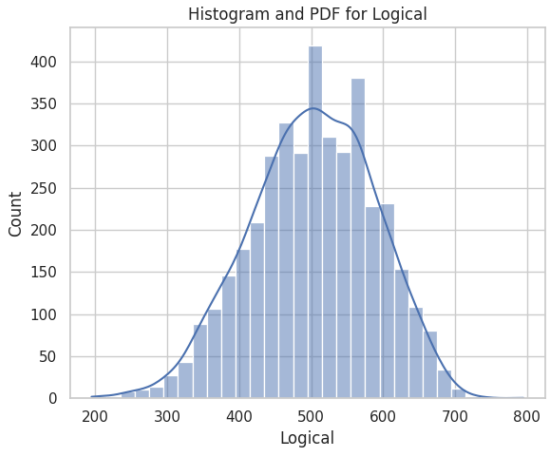

Probability Density Function for CollegeCityTier
Density
CollegeCityTier

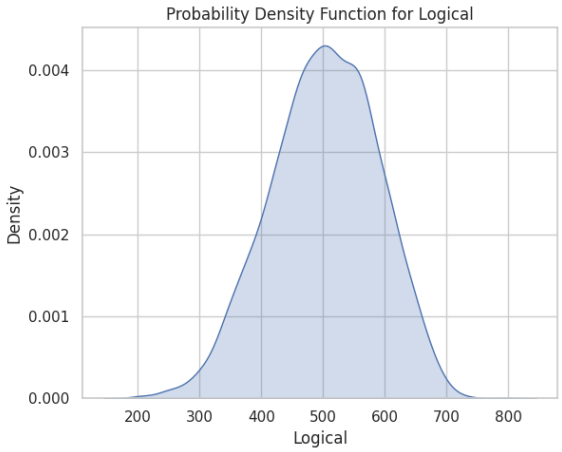

Histogram and PDF for GraduationYear
Count
GraduationYear
0
2000
4000
6000
8000
10000
12000
14000
0
250
500
750
1000
1250
1500
1750
2000

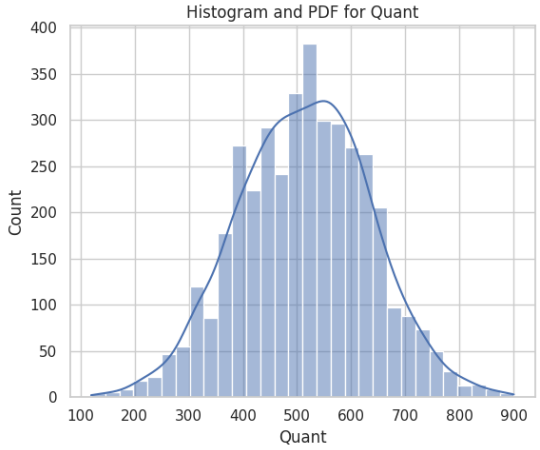

Probability Density Function for GraduationYear
Density
GraduationYear
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
500
1000
1500
2000

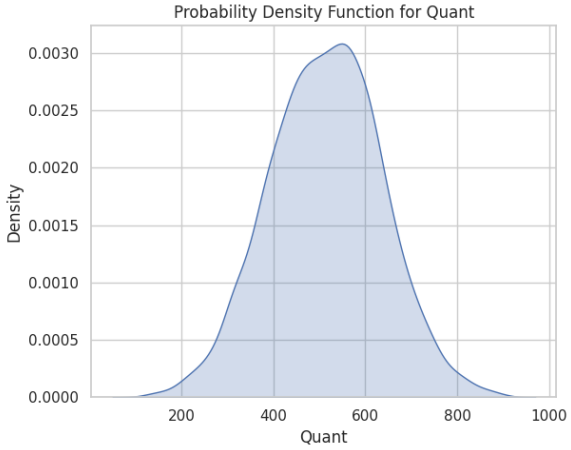

Histogram and PDF for English
Count
English
0
50
100
150
200
250
300
350
400
200
300
400
500
600
700
800
900

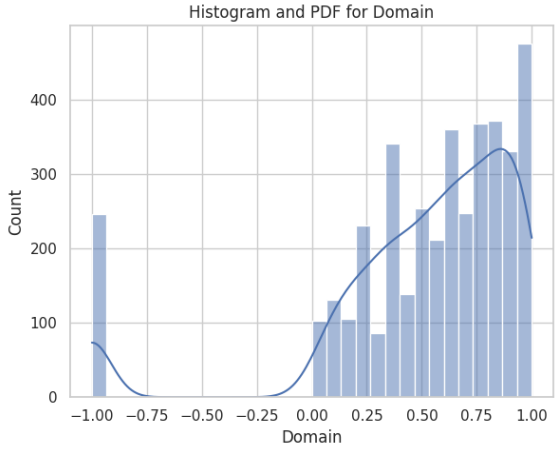

Probability Density Function for English
Density
English
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
100
200
300
400
500
600
700
800
900

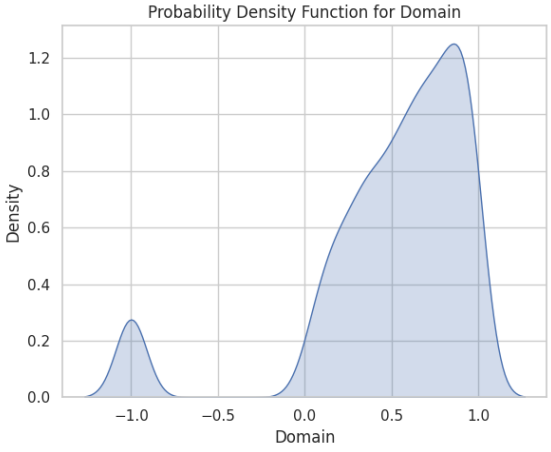

Histogram and PDF for Logical
Count
Logical
0
50
100
150
200
250
300
350
400
200
300
400
500
600
700
800

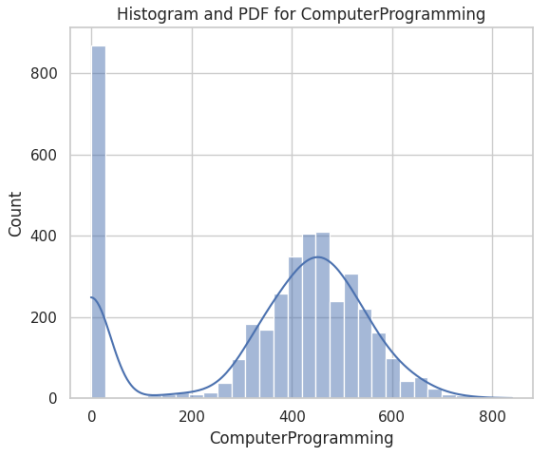

Probability Density Function for Logical
Density
Logical
0.000
0.001
0.002
0.003
0.004
200
300
400
500
600
700
800

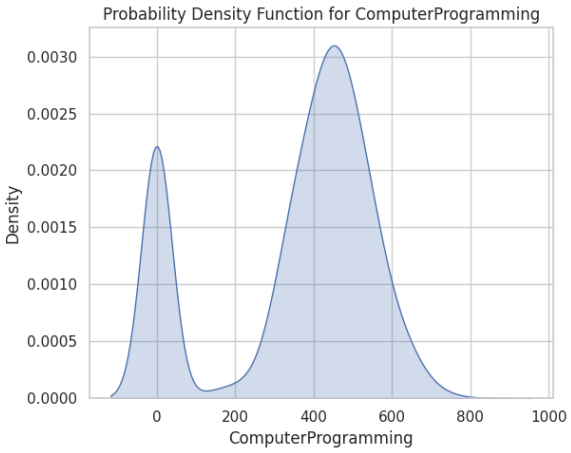

Histogram and PDF for Quant
Count
Quant

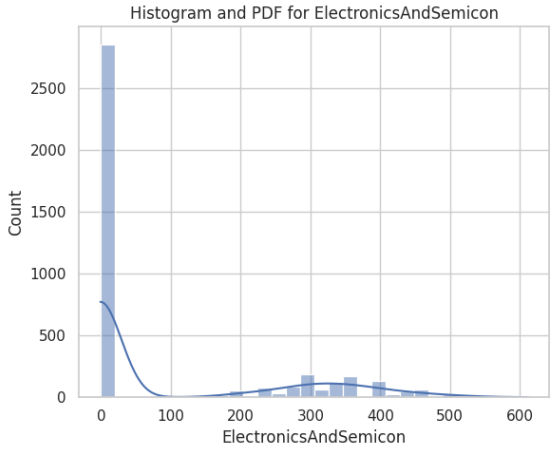

Probability Density Function for Quant
Density
Quant

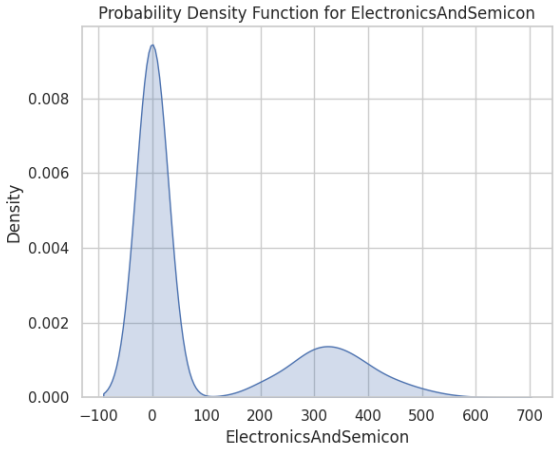

Histogram and PDF for Domain
Count
Domain

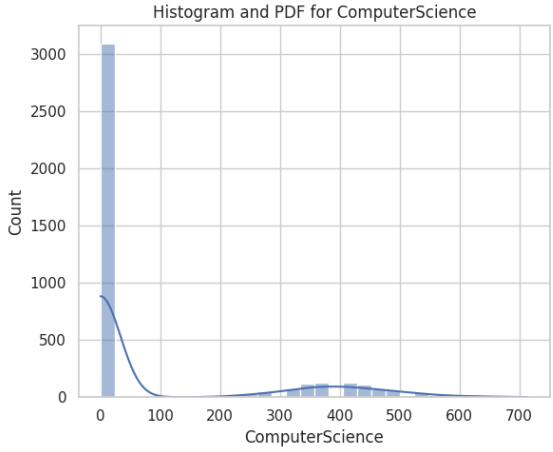

Probability Density Function for Domain
Density
Domain

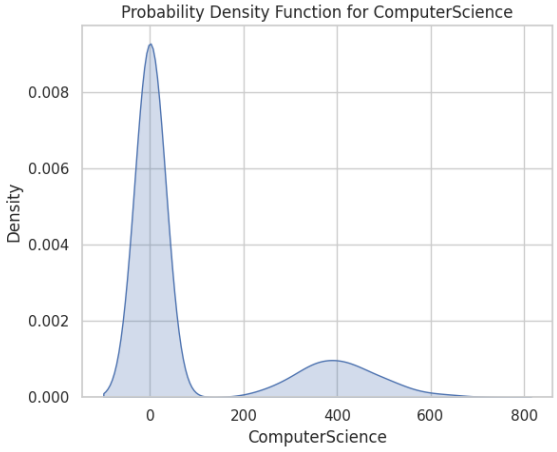

Histogram and PDF for ComputerProgramming
Count
ComputerProgramming

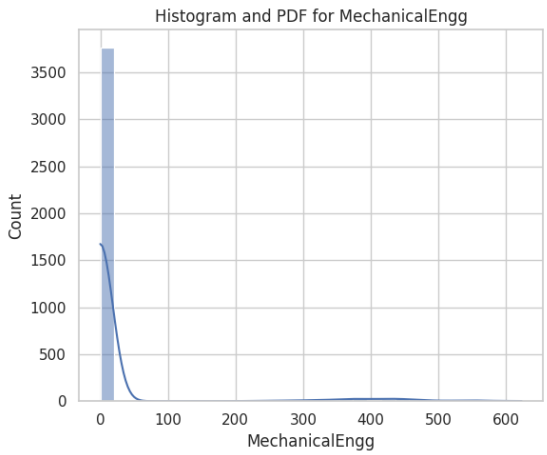

Probability Density Function for ComputerProgramming
Density
ComputerProgramming

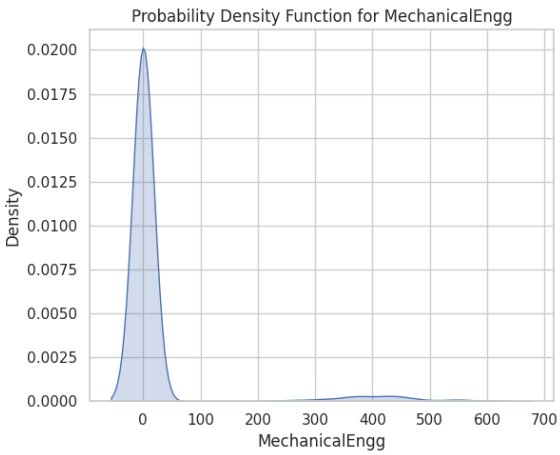

Histogram and PDF for ElectronicsAndSemicon
Count
ElectronicsAndSemicon

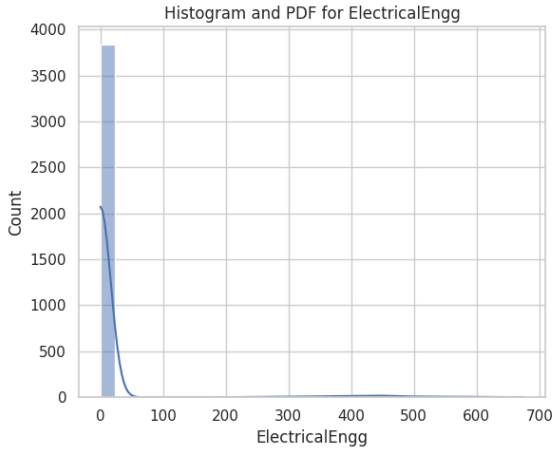

Probability Density Function for ElectronicsAndSemicon
Density
ElectronicsAndSemicon

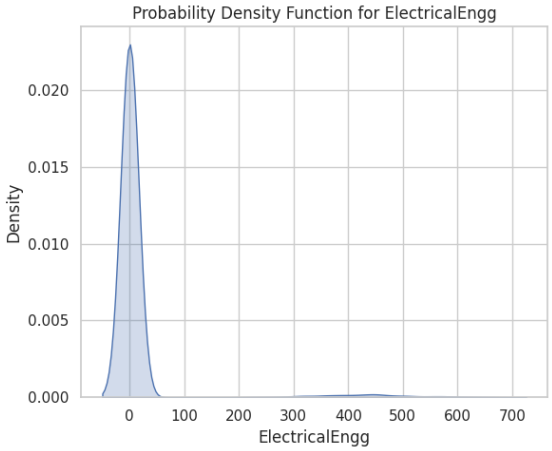

Histogram and PDF for ComputerScience
Count
ComputerScience

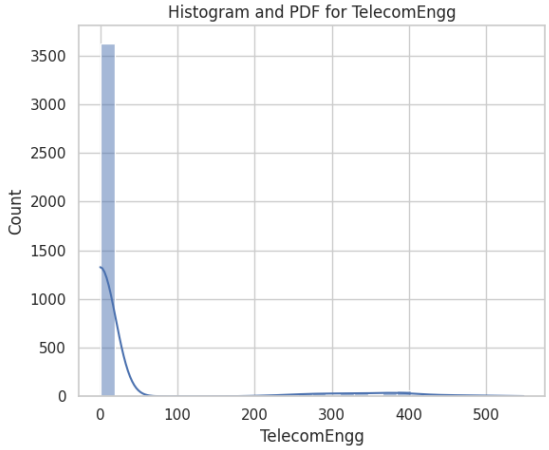

Probability Density Function for ComputerScience
Density
ComputerScience

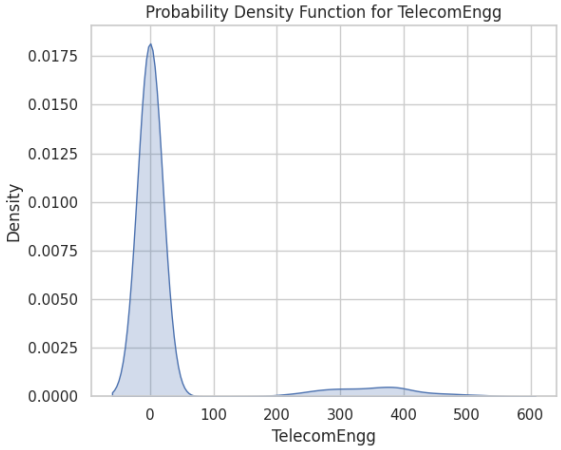

Histogram and PDF for MechanicalEngg
Count
MechanicalEngg

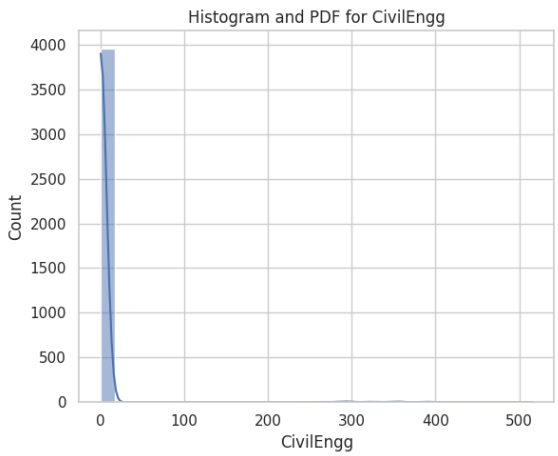

Probability Density Function for MechanicalEngg
Density
MechanicalEngg

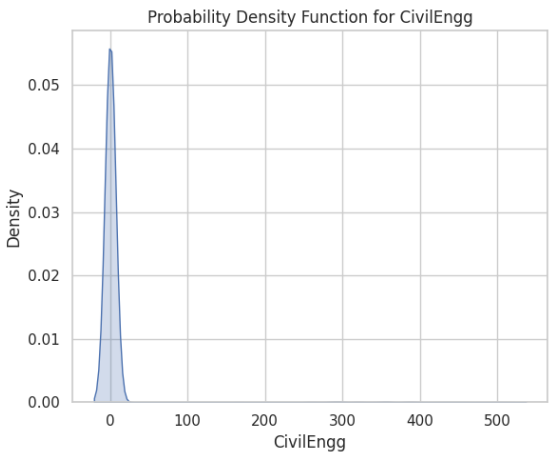

Histogram and PDF for ElectricalEngg
Count
ElectricalEngg

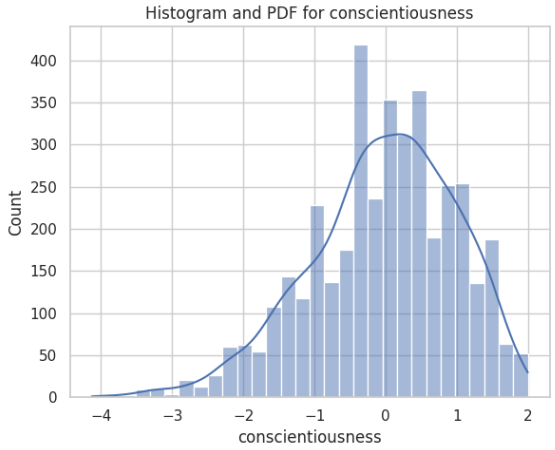

Probability Density Function for ElectricalEngg
Density
ElectricalEngg

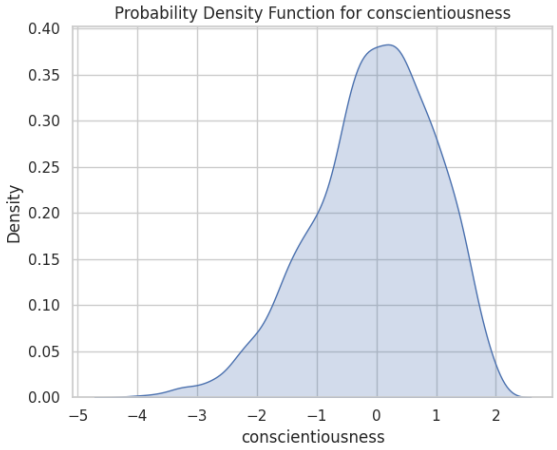

Histogram and PDF for TelecomEngg
Count
TelecomEngg

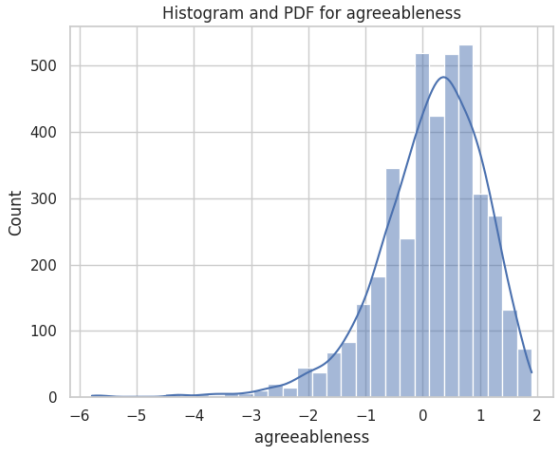

Probability Density Function for TelecomEngg
Density
TelecomEngg

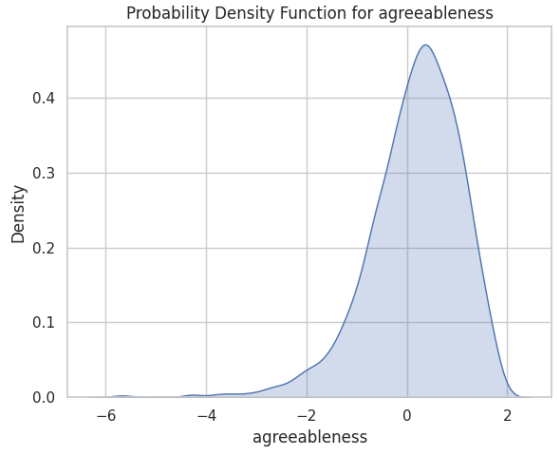

Histogram and PDF for CivilEngg
Count
CivilEngg

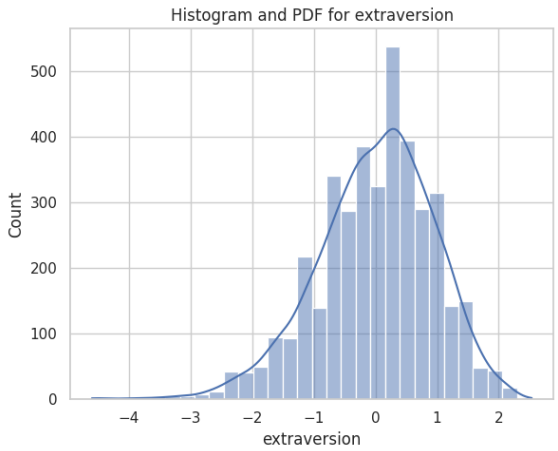

Probability Density Function for CivilEngg
Density
CivilEngg

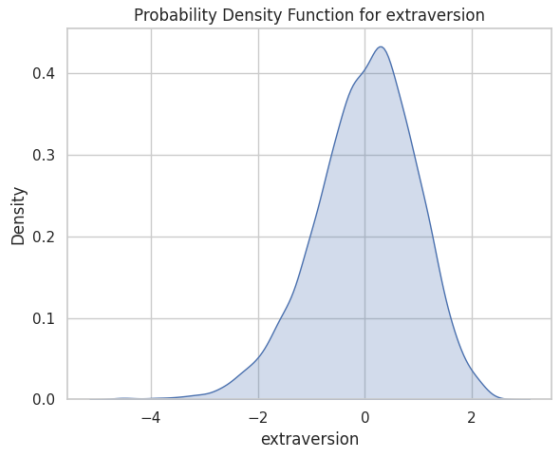

Histogram and PDF for conscientiousness
Count
conscientiousness

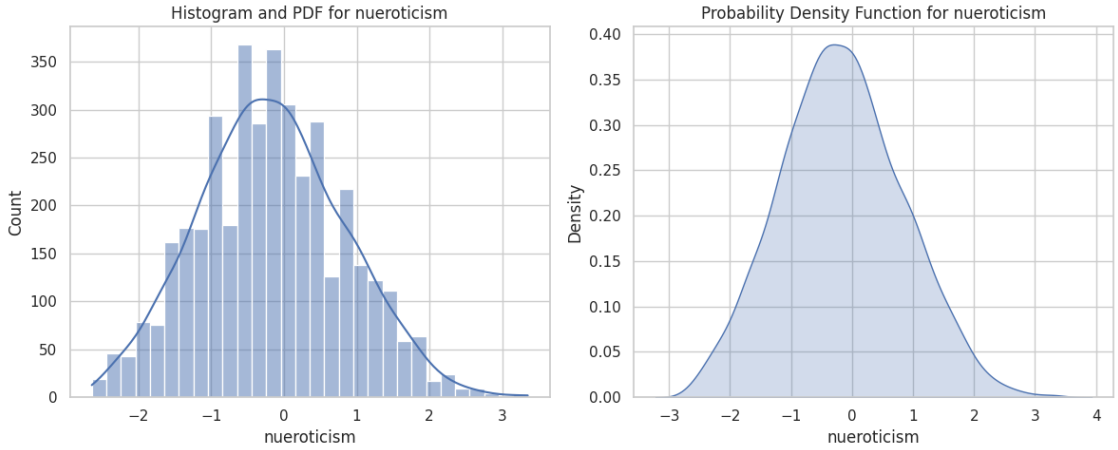

Probability Density Function for conscientiousness
Density
conscientiousness

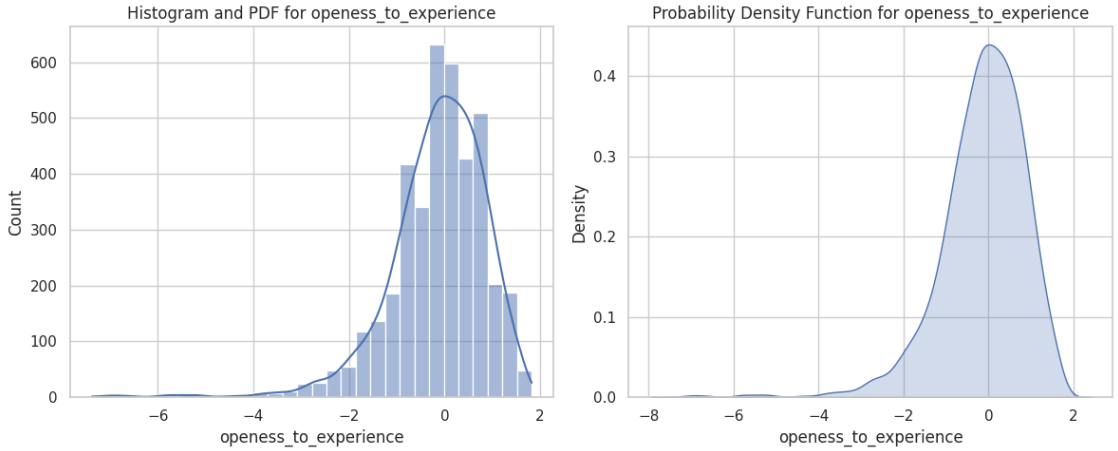

Histogram and PDF for agreeableness
Count
agreeableness


Probability Density Function for agreeableness
Density
agreeableness

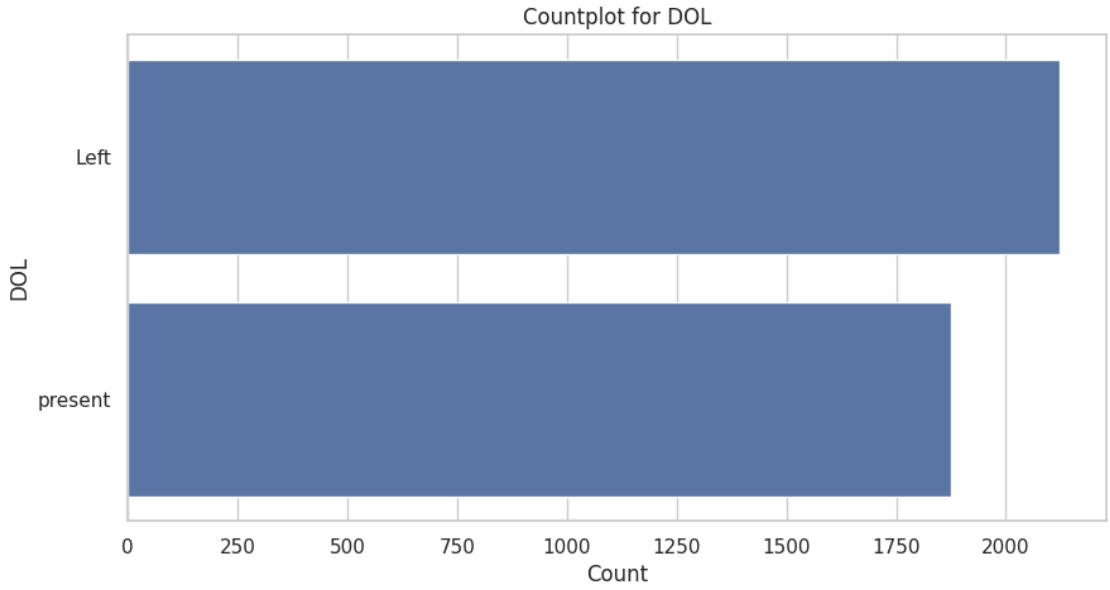

Histogram and PDF for extraversion
Count
extraversion

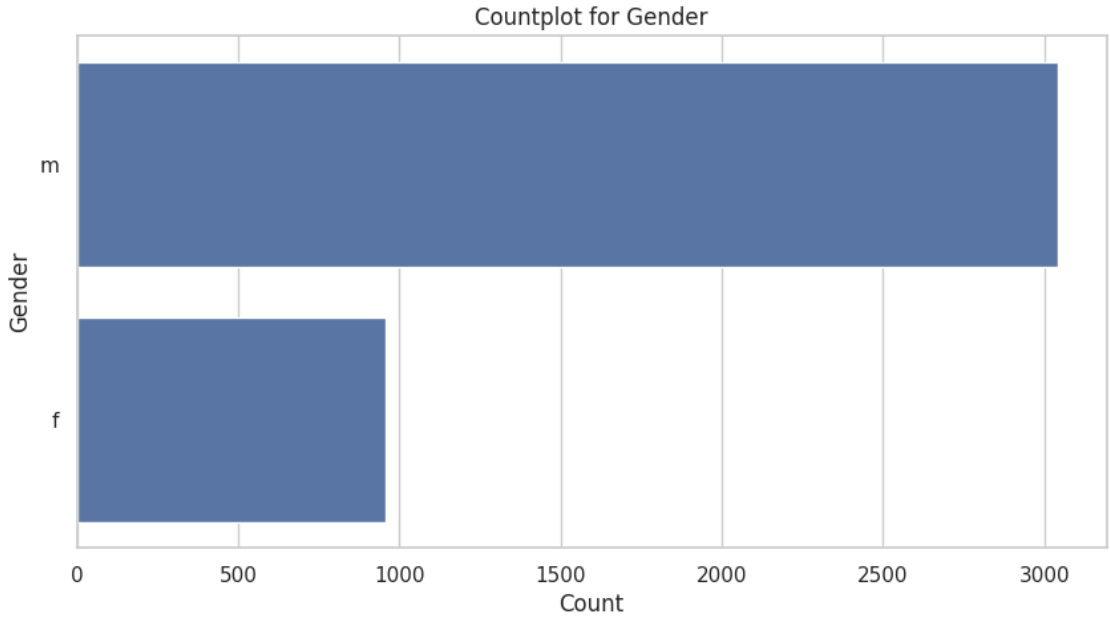

Probability Density Function for extraversion
Density
extraversion

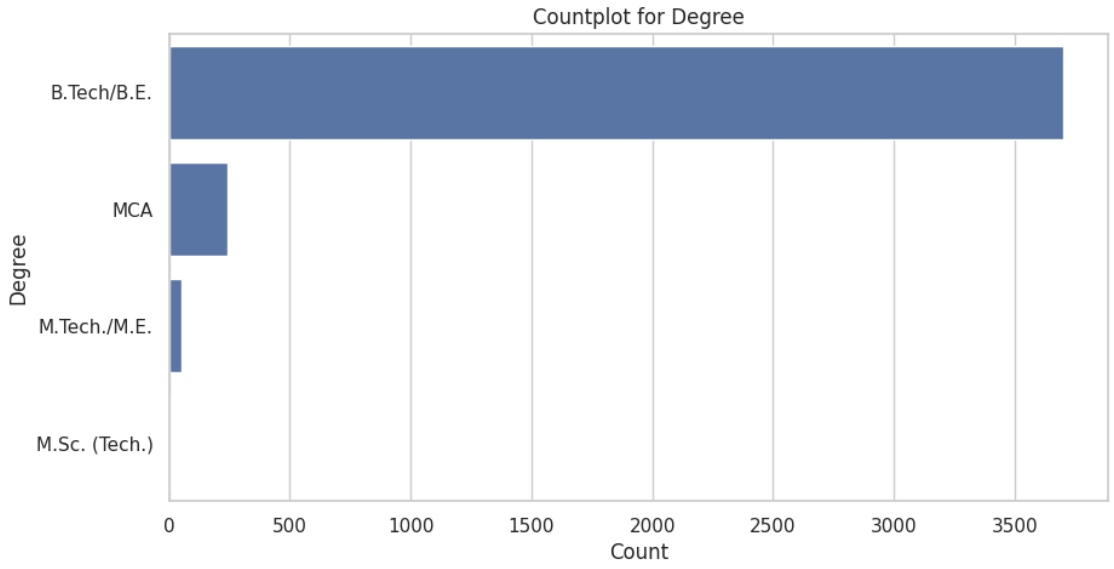


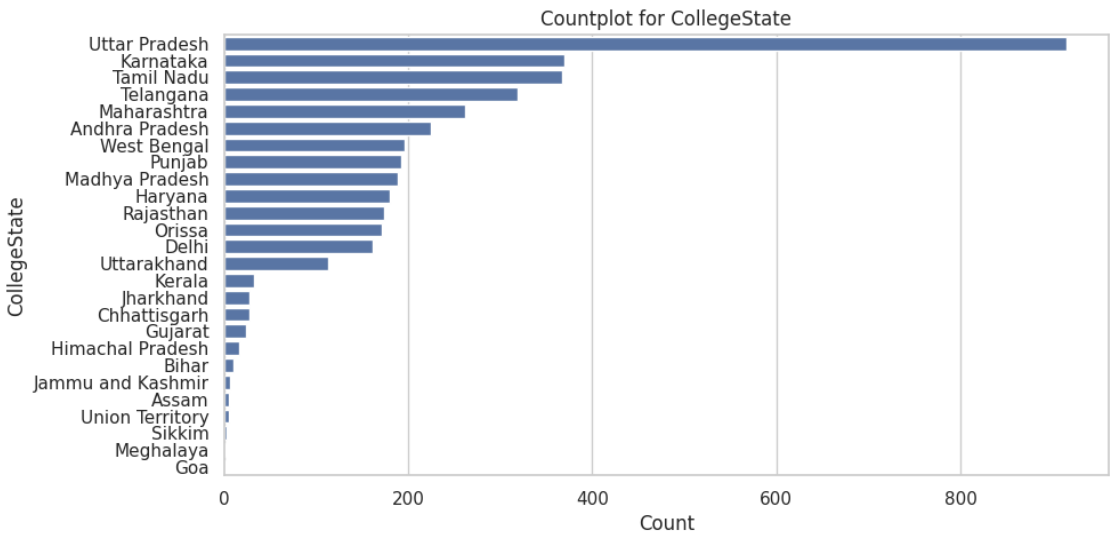


[67]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Specify the categorical columns to analyze
categorical_columns = ['DOL', 'Gender', 'Degree', 'CollegeState']

# Create countplots for each specified categorical column
for col in categorical_columns:
    plt.figure(figsize=(10, 5))
    sns.countplot(y=df[col], order=df[col].value_counts().index)
    plt.title(f'Countplot for {col}')
    plt.xlabel('Count')
    plt.ylabel(col)
    plt.show()
```


Countplot for DOL
DOL
Left
present
0
250
500
750
1000
1250
1500
1750
2000
Count

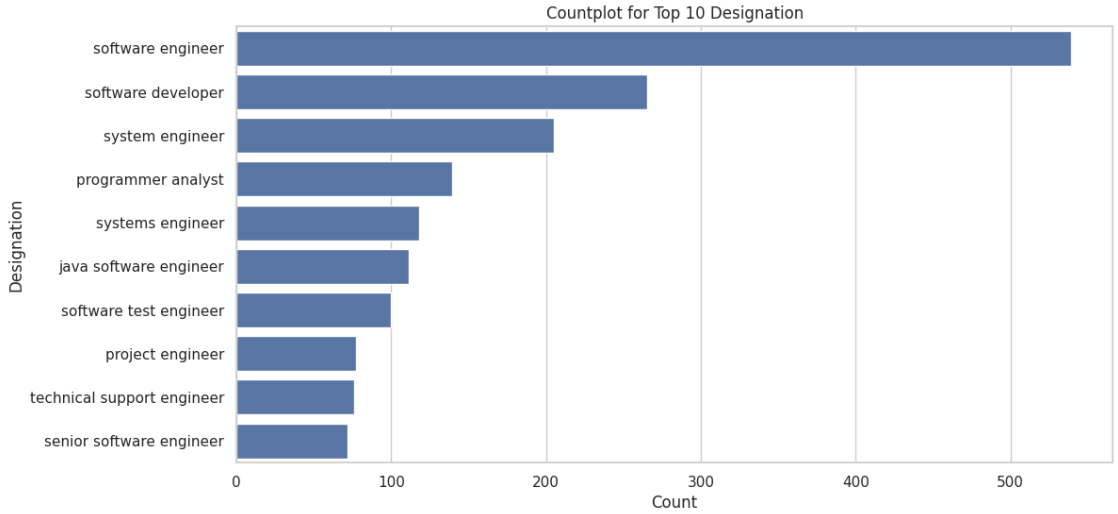

Countplot for Gender
Gender
m
f
0
500
1000
1500
2000
2500
3000
Count

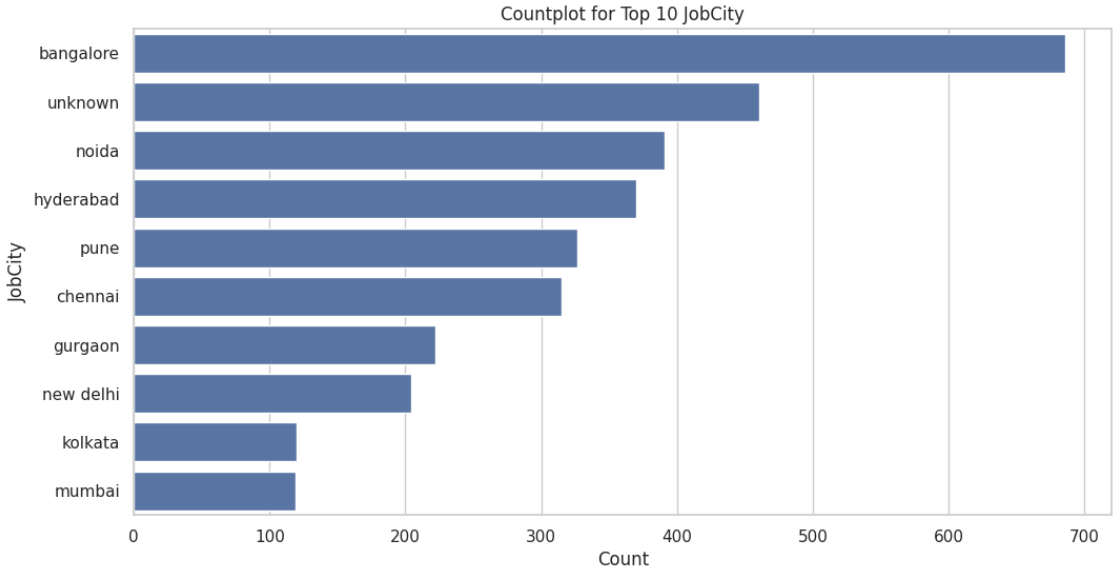


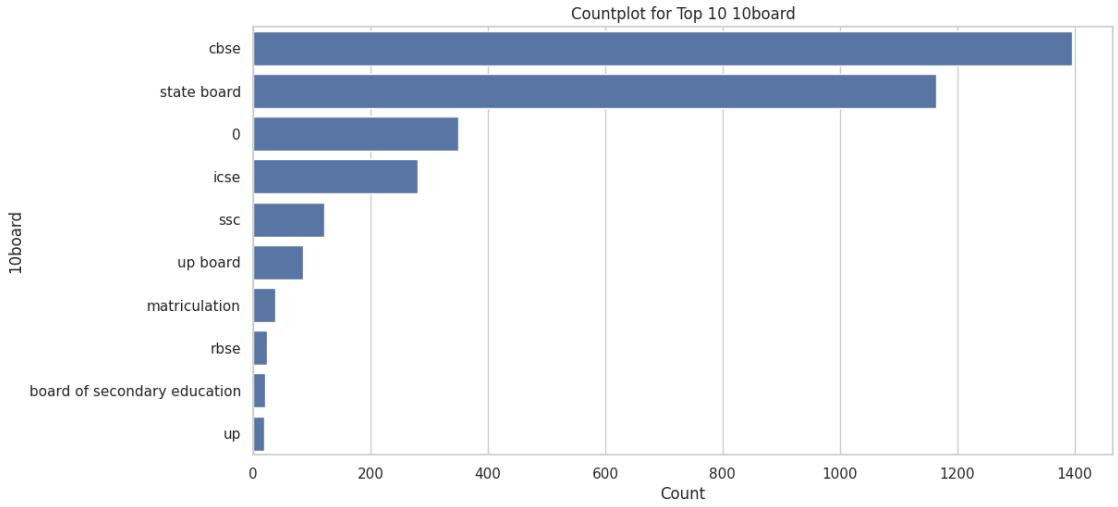


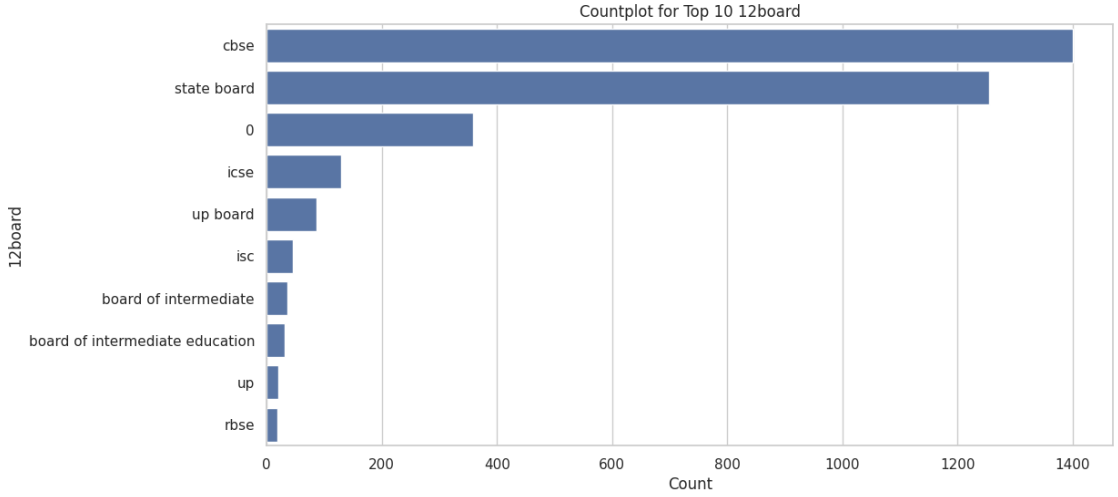
[68]:
```
# List of categorical columns to analyze
categorical_columns = ['Designation', 'JobCity', '10board', '12board',␣
 ↪'Specialization']

# Create countplots for each categorical column, limited to top 10
for col in categorical_columns:
    plt.figure(figsize=(12, 6))

    # Get top 10 categories
    top_categories = df[col].value_counts().nlargest(10).index
```

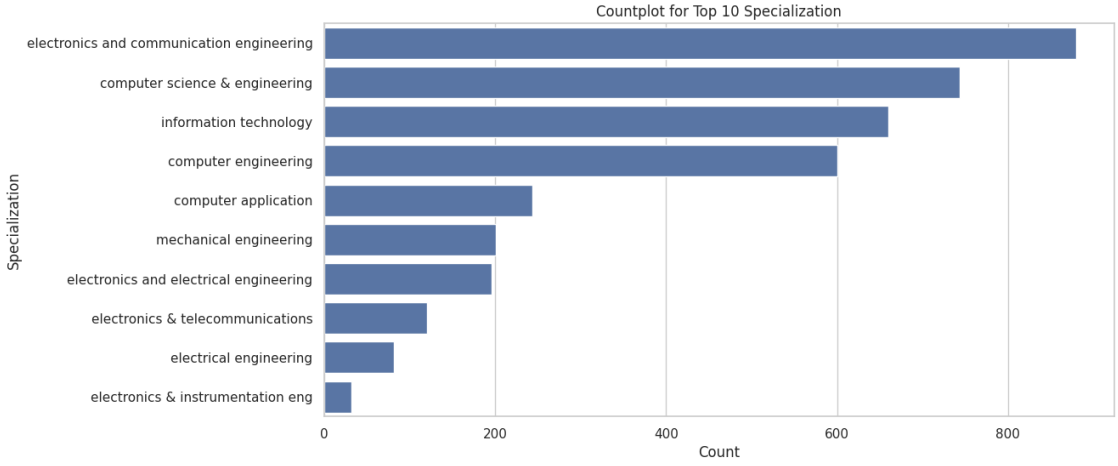
```
# Create countplot for top 10 categories
sns.countplot(y=df[col][df[col].isin(top_categories)], order=top_categories)
plt.title(f'Countplot for Top 10 {col}')
plt.xlabel('Count')
plt.ylabel(col)
plt.show()
```






Countplot for Top 10 10board
cbse
state board
0
icse
ssc
up board
matriculation
rbse
board of secondary education
up
10board
0
200
400
600
800
1000
1200
1400
Count


Countplot for Top 10 12board
cbse
state board
0
icse
up board
isc
board of intermediate
board of intermediate education
up
rbse
12board
0
200
400
600
800
1000
1200
1400
Count

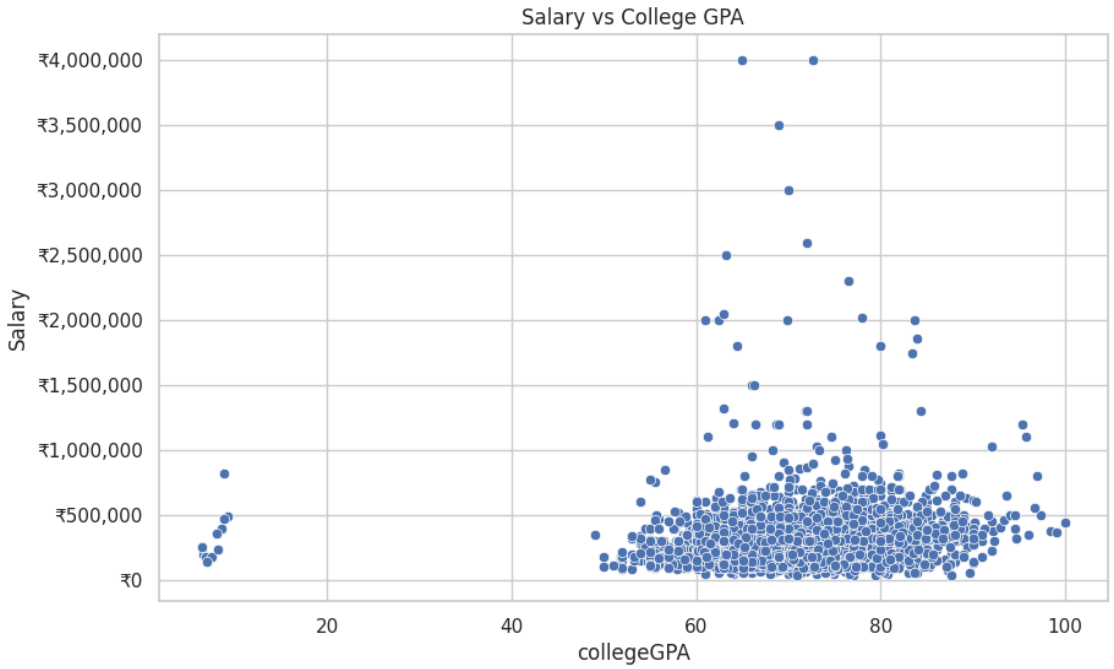

Countplot for Top 10 Specialization
electronics and communication engineering
computer science & engineering
information technology
computer engineering
computer application
mechanical engineering
electronics and electrical engineering
electronics & telecommunications
electrical engineering
electronics & instrumentation eng
Specialization
0
200
400
600
800
Count

8 Step - 4

[69]:
```
df['Salary'].value_counts()
```

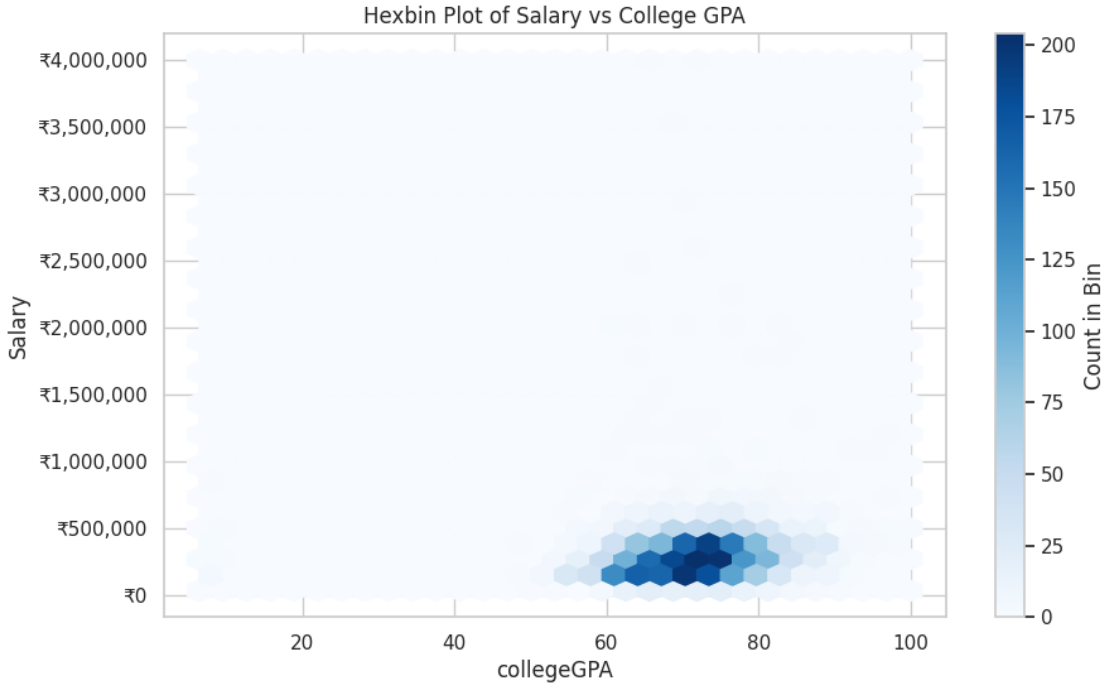
[69]:
```
Salary
300000.0     293
180000.0     239
200000.0     205
325000.0     188
120000.0     165
             …
2050000.0      1
144000.0       1
1320000.0      1
755000.0       1
925000.0       1
Name: count, Length: 177, dtype: int64
```

[70]:
```
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Set up the currency formatter
formatter = FuncFormatter(lambda x, _: f'{int(x):,}')

# Define the figure
plt.figure(figsize=(10, 6))

# Create the scatter plot
sns.scatterplot(data=df, x='collegeGPA', y='Salary')

# Title and labels
plt.title('Salary vs College GPA')
plt.xlabel('collegeGPA')
plt.ylabel('Salary')

# Add grid for better visibility
plt.grid(True)

# Apply the formatter to the y-axis
plt.gca().yaxis.set_major_formatter(formatter)

# Display the plot
plt.show()
```



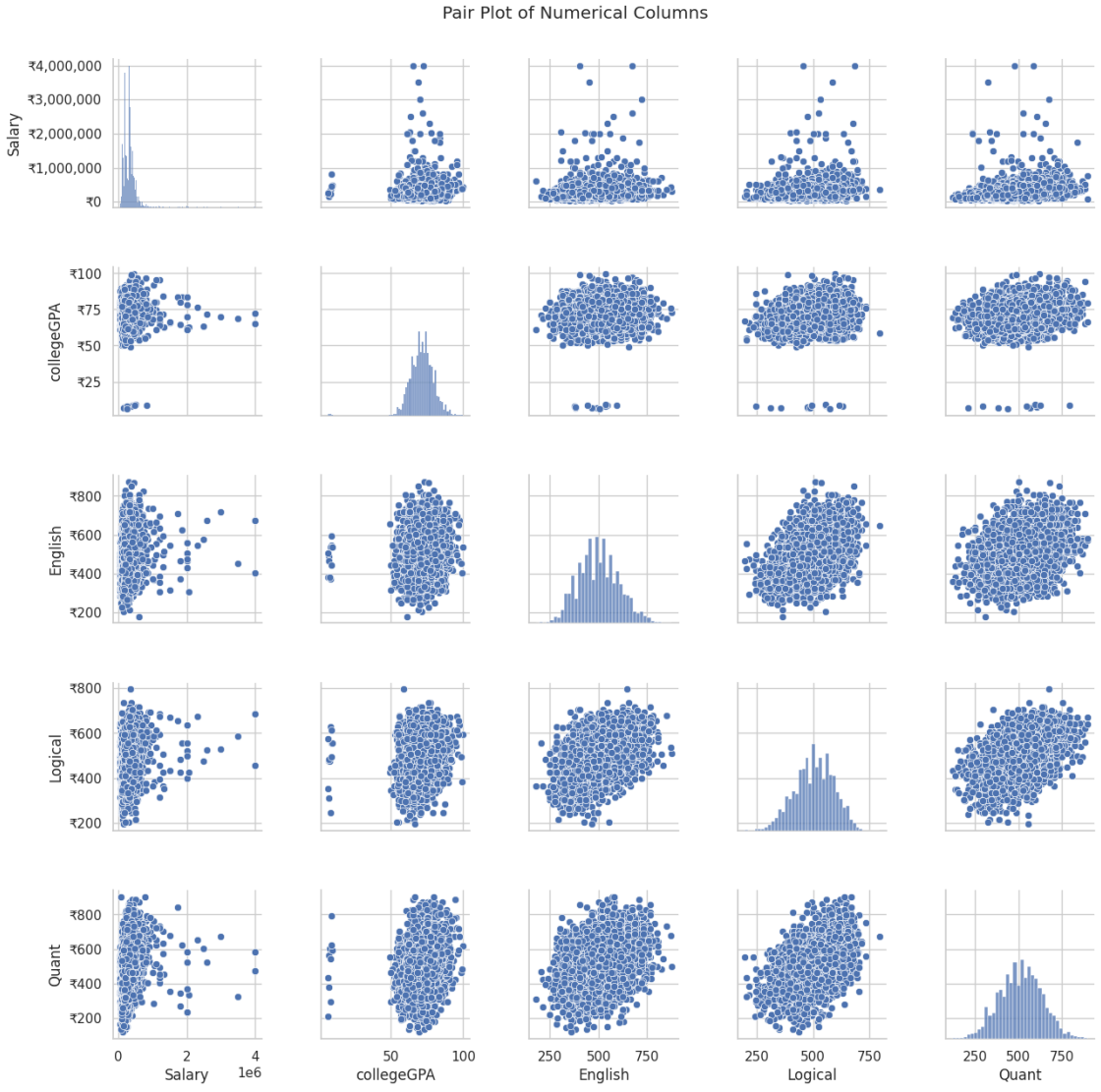
[71]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Function to format y-axis labels as currency
def currency(x, _):
    return f' {int(x):,}'

# Set up the figure
plt.figure(figsize=(10, 6))

# Create the hexbin plot
hexbin = plt.hexbin(df['collegeGPA'], df['Salary'], gridsize=30, cmap='Blues')

# Add color bar
plt.colorbar(hexbin, label='Count in Bin')

# Title and labels
plt.title('Hexbin Plot of Salary vs College GPA')
plt.xlabel('collegeGPA')
plt.ylabel('Salary')

# Apply the currency formatter to the y-axis
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))
```

```
# Show the plot
plt.show()
```



[72]:
```
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Define the numerical columns to analyze
numerical_columns = ['Salary', 'collegeGPA', 'English', 'Logical', 'Quant']

# Set the style for the plots
sns.set(style="whitegrid")

# Create the pair plot
pair_plot = sns.pairplot(df[numerical_columns])

# Set the title for the pair plot
plt.suptitle('Pair Plot of Numerical Columns', y=1.02)

# Adjust the spacing between subplots
plt.subplots_adjust(hspace=0.4, wspace=0.4)

# Apply currency formatting to y-axis labels
for ax in pair_plot.axes.flatten():
```

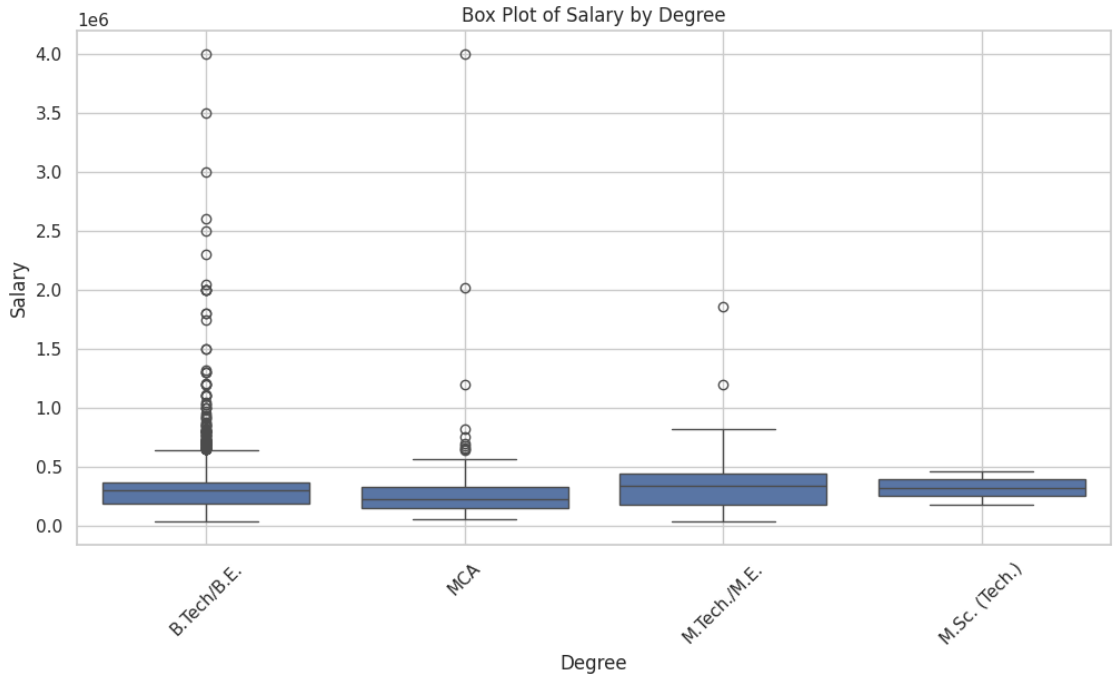
```
    ax.yaxis.set_major_formatter(FuncFormatter(currency))

# Show the plot
plt.show()
```



[73]:
```
# Create the box plot for all degrees
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='Degree', y='Salary')
plt.title('Box Plot of Salary by Degree')
plt.xlabel('Degree')
plt.ylabel('Salary')
plt.xticks(rotation=45)
plt.grid(True)
```

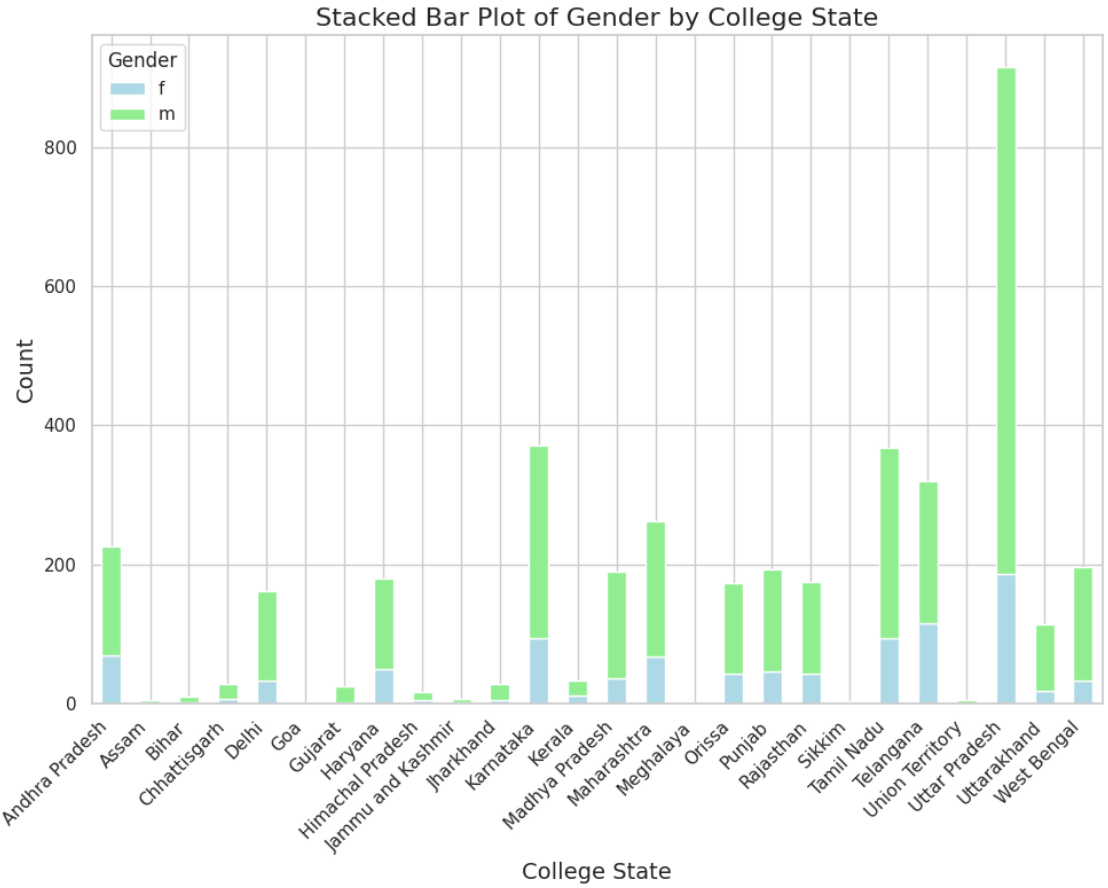
```
plt.show()
```

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



[74]:
```
# Create a pivot table for salary counts by college state and gender
pivot_table = df.pivot_table(index='CollegeState', columns='Gender',␣
 ↪values='Salary', aggfunc='count', fill_value=0)

# Set the figure size
plt.figure(figsize=(10, 8))

# Plot the stacked bar chart
pivot_table.plot(kind='bar', stacked=True, ax=plt.gca(), color=['lightblue',␣
 ↪'lightgreen'])

# Title and labels
plt.title('Stacked Bar Plot of Gender by College State', fontsize=16)
plt.xlabel('College State', fontsize=14)
plt.ylabel('Count', fontsize=14)

# Rotate x-axis labels for better visibility
plt.xticks(rotation=45, ha='right')
```

```
# Adjust layout to prevent clipping
plt.tight_layout()

# Show the plot
plt.show()
```

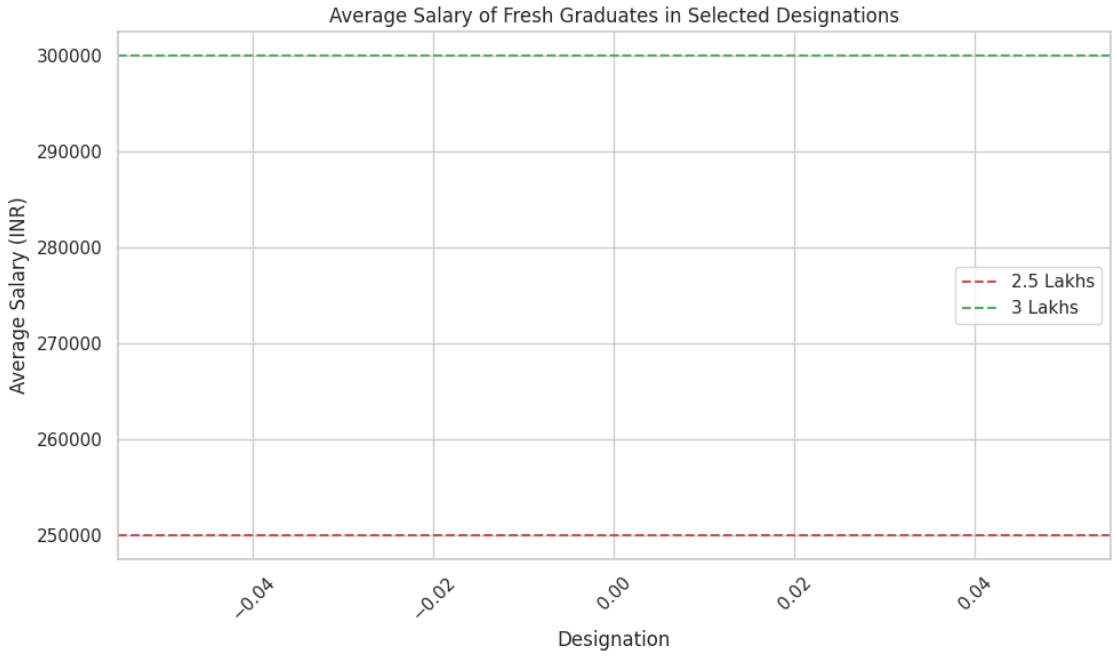


9 step 5

[75]:
```
# Define the relevant designations
designations = ['Programming Analyst', 'Software Engineer', 'Hardware␣
↪Engineer', 'Associate Engineer']

# Filter the DataFrame for these designations
filtered_df = df[df['Designation'].isin(designations)]

# Calculate average salary for each designation
```

```
average_salaries = filtered_df.groupby('Designation')['Salary'].mean().
 ↪reset_index()

# Check if the average salaries fall within the claimed range
average_salaries['Within Range'] = average_salaries['Salary'].between(250000,␣
 ↪300000)

# Print the average salaries
print(average_salaries)

# Plot the average salaries
plt.figure(figsize=(10, 6))
plt.bar(average_salaries['Designation'], average_salaries['Salary'],␣
 ↪color='skyblue')
plt.axhline(y=250000, color='r', linestyle='--', label='2.5 Lakhs')
plt.axhline(y=300000, color='g', linestyle='--', label='3 Lakhs')
plt.title('Average Salary of Fresh Graduates in Selected Designations')
plt.xlabel('Designation')
plt.ylabel('Average Salary (INR)')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

```
Empty DataFrame
Columns: [Designation, Salary, Within Range]
Index: []
```



[76]:
```
import pandas as pd
from scipy.stats import chi2_contingency

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

# Display the contingency table
print("Contingency Table:")
print(contingency_table)

# Perform the Chi-Square test
chi2, p, _, _ = chi2_contingency(contingency_table)

# Display the results
print("\nChi-Square Test Results:")
print(f"Chi-Squared Statistic: {chi2:.2f}")
print(f"P-Value: {p:.4f}")

# Conclusion based on the p-value
if p < 0.05:
    print("\nConclusion: There is a significant relationship between gender and␣
 ↪specialization.")
else:
    print("\nConclusion: There is no significant relationship between gender␣
 ↪and specialization.")
```

```
Contingency Table:
Specialization  aeronautical engineering  \
Gender
f                                       1
m                                       2

Specialization  applied electronics and instrumentation  \
Gender
f                                                    2
m                                                    7

Specialization  automobile/automotive engineering  biomedical engineering  \
Gender
f                                               0                       2
m                                               5                       0

Specialization  biotechnology  ceramic engineering  chemical engineering  \
Gender
f                           9                    0                     1
```

```
m                           6                   1                      8

Specialization  civil engineering  computer and communication engineering  \
Gender
f                              6                                        0
m                             23                                        1

Specialization  computer application  …  internal combustion engine  \
Gender                                 …
f                               59     …                          0
m                              185     …                          1

Specialization  mechanical & production engineering  \
Gender
f                                                 0
m                                                 1

Specialization  mechanical and automation  mechanical engineering  \
Gender
f                                       0                      10
m                                       5                     191

Specialization  mechatronics  metallurgical engineering  other  \
Gender
f                           1                          0      0
m                           3                          2     13

Specialization  polymer technology  power systems and automation  \
Gender
f                               0                              0
m                               1                              1

Specialization  telecommunication engineering
Gender
f                                          1
m                                          5

[2 rows x 46 columns]

Chi-Square Test Results:
Chi-Squared Statistic: 104.47
P-Value: 0.0000

Conclusion: There is a significant relationship between gender and
specialization.
```

10 Conlusion

Overall Gender Distribution: The plot shows a slight skew towards male students in most states. This is particularly evident in states like Maharashtra, Tamil Nadu, and West Bengal. However, there are also states like Goa, Himachal Pradesh, and Sikkim where the number of female students is relatively higher.

State-Specific Variations: The gender distribution varies significantly across different states. For instance, states like Bihar, Chhattisgarh, and Jharkhand have a more balanced gender ratio, whereas states like Andhra Pradesh, Delhi, and Karnataka exhibit a pronounced preference for male students.

Gender Disparity in Certain States: The plot highlights the existence of gender disparity in some states. For example, in Rajasthan, the number of male students is significantly higher than female students. This indicates a need for interventions to address gender imbalances in these regions.

11 over all Conclusion

The stacked bar plot reveals that the gender distribution in Indian colleges is not uniform across all states. While there is a general trend towards a higher proportion of male students, there are also states with more balanced gender ratios. Understanding these variations is crucial for policymakers and educational institutions to implement targeted measures to promote gender equality in higher education.

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: