AMCAT Data Analysis

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import numpy as np
import warnings
warnings.filterwarnings("ignore")

from scipy import stats

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

```
     Unnamed: 0       ID     Salary           DOJ           DOL  \
0         train   203097   420000.0   6/1/12 0:00       present
1         train   579905   500000.0   9/1/13 0:00       present
2         train   810601   325000.0   6/1/14 0:00       present
3         train   267447  1100000.0   7/1/11 0:00       present
4         train   343523   200000.0   3/1/14 0:00  3/1/15 0:00
...         ...      ...        ...           ...           ...
3738      train  1045712    95000.0   8/1/13 0:00  2/1/14 0:00
3739      train   852189   325000.0  12/1/13 0:00  4/1/15 0:00
3740      train  1240608   405000.0   1/1/15 0:00  5/1/15 0:00
3741      train   806319   400000.0   1/1/14 0:00       present
3742      train   629725   100000.0   5/1/13 0:00  4/1/14 0:00

                    Designation    JobCity Gender           DOB
10percentage  \
0      senior quality engineer  Bangalore      f  2/19/90 0:00
84.30
1            assistant manager     Indore      m  10/4/89 0:00
85.40
2             systems engineer    Chennai      f   8/3/92 0:00
85.00
3     senior software engineer    Gurgaon      m  12/5/89 0:00
85.60
4                          get    Manesar      m  2/27/91 0:00
78.00
...                        ...        ...    ...           ...
...
3738         software engineer    Gurgaon      m   7/1/90 0:00
65.17
3739    java software engineer         -1      m  3/26/91 0:00
71.85
3740        database developer         -1      m  7/25/92 0:00
84.00
```

```
3741        software developer      Noida      m  3/16/89 0:00
65.40
3742       project coordinator    Chennai      m  3/15/91 0:00
86.00

      ... ComputerScience  MechanicalEngg  ElectricalEngg TelecomEngg
\
0     ...            -1.0            -1.0            -1.0        -1.0

1     ...            -1.0            -1.0            -1.0        -1.0

2     ...            -1.0            -1.0            -1.0        -1.0

3     ...            -1.0            -1.0            -1.0        -1.0

4     ...            -1.0            -1.0            -1.0        -1.0

...   ...             ...             ...             ...         ...

3738  ...           500.0            -1.0            -1.0        -1.0

3739  ...            -1.0            -1.0            -1.0        -1.0

3740  ...            -1.0            -1.0            -1.0        -1.0

3741  ...            -1.0            -1.0            -1.0        -1.0

3742  ...             NaN             NaN             NaN         NaN


      CivilEngg  conscientiousness agreeableness extraversion
nueroticism  \
0          -1.0             0.9737        0.8128       0.5269
1.35490
1          -1.0            -0.7335        0.3789       1.2396     -
0.10760
2          -1.0             0.2718        1.7109       0.1637     -
0.86820
3          -1.0             0.0464        0.3448      -0.3440     -
0.40780
4          -1.0            -0.8810       -0.2793      -1.0697
0.09163
...         ...                ...           ...          ...
...
3738       -1.0            -0.4463       -0.2871       0.4711     -
1.62890
3739       -1.0             0.4155       -0.1206       0.1637     -
0.36120
3740       -1.0            -1.1644       -1.1196      -0.1437
0.52620
```

```
3741       -1.0              0.2718         0.0459       0.7785     -
0.61470
3742        NaN                NaN            NaN          NaN
NaN

      openess_to_experience
0                   -0.4455
1                    0.8637
2                    0.6721
3                   -0.9194
4                   -0.1295
...                     ...
3738                 0.0973
3739                 0.2889
3740                -1.6273
3741                -1.8189
3742                    NaN

[3743 rows x 39 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3743 entries, 0 to 3742
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             3743 non-null   object
 1   ID                     3743 non-null   int64
 2   Salary                 3743 non-null   float64
 3   DOJ                    3743 non-null   object
 4   DOL                    3743 non-null   object
 5   Designation            3743 non-null   object
 6   JobCity                3743 non-null   object
 7   Gender                 3743 non-null   object
 8   DOB                    3743 non-null   object
 9   10percentage           3743 non-null   float64
 10  10board                3743 non-null   object
 11  12graduation           3743 non-null   int64
 12  12percentage           3743 non-null   float64
 13  12board                3743 non-null   object
 14  CollegeID              3742 non-null   float64
 15  CollegeTier            3742 non-null   float64
 16  Degree                 3742 non-null   object
 17  Specialization         3742 non-null   object
 18  collegeGPA             3742 non-null   float64
 19  CollegeCityID          3742 non-null   float64
 20  CollegeCityTier        3742 non-null   float64
 21  CollegeState           3742 non-null   object
 22  GraduationYear         3742 non-null   float64
```

```
 23  English                3742 non-null   float64
 24  Logical                3742 non-null   float64
 25  Quant                  3742 non-null   float64
 26  Domain                 3742 non-null   float64
 27  ComputerProgramming    3742 non-null   float64
 28  ElectronicsAndSemicon  3742 non-null   float64
 29  ComputerScience        3742 non-null   float64
 30  MechanicalEngg         3742 non-null   float64
 31  ElectricalEngg         3742 non-null   float64
 32  TelecomEngg            3742 non-null   float64
 33  CivilEngg              3742 non-null   float64
 34  conscientiousness      3742 non-null   float64
 35  agreeableness          3742 non-null   float64
 36  extraversion           3742 non-null   float64
 37  nueroticism            3742 non-null   float64
 38  openess_to_experience  3742 non-null   float64
dtypes: float64(25), int64(2), object(12)
memory usage: 1.1+ MB
```

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

```
df.columns
```

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

```
df.columns = df.columns.str.lower()
```

```
df.head()
```

```
       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  ...             -1.0            -
1.0
1                            cbse  ...             -1.0            -
1.0
2                            cbse  ...             -1.0            -
1.0
3                            cbse  ...             -1.0            -
1.0
4                            cbse  ...             -1.0            -
1.0

  electricalengg  telecomengg  civilengg conscientiousness
agreeableness  \
0           -1.0         -1.0       -1.0            0.9737
0.8128
1           -1.0         -1.0       -1.0           -0.7335
0.3789
2           -1.0         -1.0       -1.0            0.2718
1.7109
3           -1.0         -1.0       -1.0            0.0464
0.3448
4           -1.0         -1.0       -1.0           -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]
```

```
df['doj'] = pd.to_datetime(df['doj'])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3743 entries, 0 to 3742
Data columns (total 38 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   id                      3743 non-null   int64
 1   salary                  3743 non-null   float64
 2   doj                     3743 non-null   datetime64[ns]
 3   dol                     3743 non-null   object
 4   designation             3743 non-null   object
 5   jobcity                 3743 non-null   object
 6   gender                  3743 non-null   object
 7   dob                     3743 non-null   object
 8   10percentage            3743 non-null   float64
 9   10board                 3743 non-null   object
 10  12graduation            3743 non-null   int64
 11  12percentage            3743 non-null   float64
 12  12board                 3743 non-null   object
 13  collegeid               3742 non-null   float64
 14  collegetier             3742 non-null   float64
 15  degree                  3742 non-null   object
 16  specialization          3742 non-null   object
 17  collegegpa              3742 non-null   float64
 18  collegecityid           3742 non-null   float64
 19  collegecitytier         3742 non-null   float64
 20  collegestate            3742 non-null   object
 21  graduationyear          3742 non-null   float64
 22  english                 3742 non-null   float64
 23  logical                 3742 non-null   float64
 24  quant                   3742 non-null   float64
 25  domain                  3742 non-null   float64
 26  computerprogramming     3742 non-null   float64
 27  electronicsandsemicon   3742 non-null   float64
 28  computerscience         3742 non-null   float64
 29  mechanicalengg          3742 non-null   float64
 30  electricalengg          3742 non-null   float64
 31  telecomengg             3742 non-null   float64
 32  civilengg               3742 non-null   float64
 33  conscientiousness       3742 non-null   float64
 34  agreeableness           3742 non-null   float64
 35  extraversion            3742 non-null   float64
 36  nueroticism             3742 non-null   float64
 37  openess_to_experience   3742 non-null   float64
dtypes: datetime64[ns](1), float64(25), int64(2), object(10)
memory usage: 1.1+ MB

df.head()

       id     salary        doj          dol                designation
\
```

```
0  203097   420000.0 2012-06-01      present   senior quality engineer

1  579905   500000.0 2013-09-01      present         assistant manager

2  810601   325000.0 2014-06-01      present          systems engineer

3  267447  1100000.0 2011-07-01      present  senior software engineer

4  343523   200000.0 2014-03-01  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  ...             -1.0            -
1.0
1                            cbse  ...             -1.0            -
1.0
2                            cbse  ...             -1.0            -
1.0
3                            cbse  ...             -1.0            -
1.0
4                            cbse  ...             -1.0            -
1.0

  electricalengg  telecomengg  civilengg conscientiousness
agreeableness  \
0           -1.0         -1.0       -1.0            0.9737
0.8128
1           -1.0         -1.0       -1.0           -0.7335
0.3789
2           -1.0         -1.0       -1.0            0.2718
1.7109
3           -1.0         -1.0       -1.0            0.0464
0.3448
4           -1.0         -1.0       -1.0           -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]
```

```
df.shape
```

```
(3743, 38)
```

```
unique_cities = df['jobcity'].unique()
unique_cities
```

```
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'], dtype=object)

df.jobcity = df.jobcity.str.strip().str.lower()

unique_cities_cleaned = df['jobcity'].unique()
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 )']
```

```
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',
    'nouda': 'Noida',
    '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': 'Tirunelveli',
    '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',
}

df['jobcity'] = df['jobcity'].replace(city_mapping)

df['jobcity'] = df.jobcity.str.strip().str.lower()

df
```

```
           id     salary        doj           dol
designation  \
0      203097   420000.0 2012-06-01      present   senior quality
engineer
1      579905   500000.0 2013-09-01      present         assistant
manager
2      810601   325000.0 2014-06-01      present          systems
engineer
3      267447  1100000.0 2011-07-01      present  senior software
engineer
4      343523   200000.0 2014-03-01  3/1/15 0:00
get
...       ...        ...        ...          ...
...
3738  1045712    95000.0 2013-08-01  2/1/14 0:00         software
engineer
3739   852189   325000.0 2013-12-01  4/1/15 0:00    java software
engineer
3740  1240608   405000.0 2015-01-01  5/1/15 0:00        database
developer
3741   806319   400000.0 2014-01-01      present        software
developer
3742   629725   100000.0 2013-05-01  4/1/14 0:00       project
coordinator

        jobcity gender           dob  10percentage  \
0     bangalore      f  2/19/90 0:00         84.30
1        indore      m  10/4/89 0:00         85.40
2       chennai      f   8/3/92 0:00         85.00
3       gurgaon      m  12/5/89 0:00         85.60
4       manesar      m  2/27/91 0:00         78.00
...         ...    ...           ...           ...
3738    gurgaon      m   7/1/90 0:00         65.17
3739    unknown      m  3/26/91 0:00         71.85
3740    unknown      m  7/25/92 0:00         84.00
3741      noida      m  3/16/89 0:00         65.40
3742    chennai      m  3/15/91 0:00         86.00

                              10board  ...  computerscience
mechanicalengg  \
0     board ofsecondary education,ap  ...             -1.0
```

```
-1.0
1                              cbse  ...             -1.0
-1.0
2                              cbse  ...             -1.0
-1.0
3                              cbse  ...             -1.0
-1.0
4                              cbse  ...             -1.0
-1.0
...                             ...  ...              ...
...
3738                     state board  ...            500.0
-1.0
3739                           icse  ...             -1.0
-1.0
3740                           cbse  ...             -1.0
-1.0
3741                           cbse  ...             -1.0
-1.0
3742                     state board  ...              NaN
NaN

     electricalengg  telecomengg  civilengg conscientiousness
agreeableness  \
0              -1.0         -1.0       -1.0            0.9737
0.8128
1              -1.0         -1.0       -1.0           -0.7335
0.3789
2              -1.0         -1.0       -1.0            0.2718
1.7109
3              -1.0         -1.0       -1.0            0.0464
0.3448
4              -1.0         -1.0       -1.0           -0.8810       -
0.2793
...             ...          ...        ...               ...
...
3738           -1.0         -1.0       -1.0           -0.4463       -
0.2871
3739           -1.0         -1.0       -1.0            0.4155       -
0.1206
3740           -1.0         -1.0       -1.0           -1.1644       -
1.1196
3741           -1.0         -1.0       -1.0            0.2718
0.0459
3742            NaN          NaN        NaN               NaN
NaN

      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
...            ...          ...                   ...
3738        0.4711     -1.62890                0.0973
3739        0.1637     -0.36120                0.2889
3740       -0.1437      0.52620               -1.6273
3741        0.7785     -0.61470               -1.8189
3742           NaN          NaN                   NaN

[3743 rows x 38 columns]
```

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

```
df
```

```
           id     salary        doj      dol                designation
\
0      203097   420000.0 2012-06-01  present   senior quality engineer

1      579905   500000.0 2013-09-01  present         assistant manager

2      810601   325000.0 2014-06-01  present          systems engineer

3      267447  1100000.0 2011-07-01  present  senior software engineer

4      343523   200000.0 2014-03-01     Left                       get

...       ...        ...        ...      ...                       ...

3738  1045712    95000.0 2013-08-01     Left         software engineer

3739   852189   325000.0 2013-12-01     Left    java software engineer

3740  1240608   405000.0 2015-01-01     Left        database developer

3741   806319   400000.0 2014-01-01  present        software developer

3742   629725   100000.0 2013-05-01     Left       project coordinator


        jobcity gender           dob  10percentage  \
0     bangalore      f  2/19/90 0:00         84.30
1        indore      m  10/4/89 0:00         85.40
2       chennai      f   8/3/92 0:00         85.00
3       gurgaon      m  12/5/89 0:00         85.60
4       manesar      m  2/27/91 0:00         78.00
...         ...    ...           ...           ...
3738    gurgaon      m   7/1/90 0:00         65.17
3739    unknown      m  3/26/91 0:00         71.85
```

```
3740    unknown      m  7/25/92 0:00         84.00
3741      noida      m  3/16/89 0:00         65.40
3742    chennai      m  3/15/91 0:00         86.00

                              10board  ...  computerscience
mechanicalengg  \
0     board ofsecondary education,ap  ...             -1.0
-1.0
1                                cbse  ...             -1.0
-1.0
2                                cbse  ...             -1.0
-1.0
3                                cbse  ...             -1.0
-1.0
4                                cbse  ...             -1.0
-1.0
...                               ...  ...              ...
...
3738                     state board  ...            500.0
-1.0
3739                             icse  ...             -1.0
-1.0
3740                             cbse  ...             -1.0
-1.0
3741                             cbse  ...             -1.0
-1.0
3742                     state board  ...              NaN
NaN

     electricalengg  telecomengg  civilengg conscientiousness
agreeableness  \
0              -1.0         -1.0       -1.0            0.9737
0.8128
1              -1.0         -1.0       -1.0           -0.7335
0.3789
2              -1.0         -1.0       -1.0            0.2718
1.7109
3              -1.0         -1.0       -1.0            0.0464
0.3448
4              -1.0         -1.0       -1.0           -0.8810       -
0.2793
...             ...          ...        ...               ...
...
3738           -1.0         -1.0       -1.0           -0.4463       -
0.2871
3739           -1.0         -1.0       -1.0            0.4155       -
0.1206
3740           -1.0         -1.0       -1.0           -1.1644       -
1.1196
```

```
3741           -1.0         -1.0       -1.0             0.2718
0.0459
3742            NaN          NaN        NaN                NaN
NaN

      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
...            ...          ...                    ...
3738        0.4711     -1.62890                 0.0973
3739        0.1637     -0.36120                 0.2889
3740       -0.1437      0.52620                -1.6273
3741        0.7785     -0.61470                -1.8189
3742           NaN          NaN                   NaN

[3743 rows x 38 columns]
```

```
df['dol'].value_counts()
```

```
dol
Left       2004
present    1739
Name: count, dtype: int64
```

```
df.salary.mean().round(2)
```

```
np.float64(308273.84)
```

```
df.salary.max()
```

```
np.float64(4000000.0)
```

```
df.salary.min()
```

```
np.float64(35000.0)
```

```
df.salary.min()
```

```
np.float64(35000.0)
```

```
df.gender.value_counts()
```

```
gender
m    2857
f     886
Name: count, dtype: int64
```

```
df.computerscience = df.computerscience.replace(-1,0)
df.mechanicalengg = df.mechanicalengg.replace(-1,0)
```

```
df.electricalengg = df.electricalengg.replace(-1,0)
df.telecomengg = df.telecomengg.replace(-1,0)
df.civilengg = df.civilengg.replace(-1,0)

df.head()
```

```
       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            0.0          0.0
0.0
1              0.0             0.0            0.0          0.0
0.0
2              0.0             0.0            0.0          0.0
0.0
3              0.0             0.0            0.0          0.0
0.0
4              0.0             0.0            0.0          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]
```

```
df['salary'].describe()
```

```
count    3.743000e+03
mean     3.082738e+05
std      2.170049e+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
```

```
pd.options.display.float_format = '{:,.0f}'.format

# Display the describe() output for the 'salary' column
df.describe().transpose()
```

	count	mean	\
id	3,743	663,448	
salary	3,743	308,274	
doj	3743	2013-06-30 04:16:59.503072768	
10percentage	3,743	78	
12graduation	3,743	2,008	
12percentage	3,743	74	
collegeid	3,742	5,120	
collegetier	3,742	2	
collegegpa	3,742	71	
collegecityid	3,742	5,120	
collegecitytier	3,742	0	
graduationyear	3,742	2,012	
english	3,742	502	
logical	3,742	502	
quant	3,742	514	
domain	3,742	1	
computerprogramming	3,742	354	
electronicsandsemicon	3,742	96	
computerscience	3,742	91	
mechanicalengg	3,742	24	
electricalengg	3,742	17	

telecomengg	3,742	33
civilengg	3,742	3
conscientiousness	3,742	-0
agreeableness	3,742	0
extraversion	3,742	0
nueroticism	3,742	-0
openess_to_experience	3,742	-0

	min	25%	\
id	11,244	334,294	
salary	35,000	180,000	
doj	1991-06-01 00:00:00	2012-09-01 00:00:00	
10percentage	43	72	
12graduation	1,995	2,007	
12percentage	40	66	
collegeid	2	494	
collegetier	1	2	
collegegpa	6	66	
collegecityid	2	494	
collegecitytier	0	0	
graduationyear	0	2,012	
english	180	425	
logical	195	445	
quant	120	430	
domain	-1	0	
computerprogramming	-1	295	
electronicsandsemicon	-1	-1	
computerscience	0	0	
mechanicalengg	0	0	
electricalengg	0	0	
telecomengg	0	0	
civilengg	0	0	
conscientiousness	-4	-1	
agreeableness	-6	-0	
extraversion	-5	-1	
nueroticism	-3	-1	
openess_to_experience	-7	-1	

	50%	75%	\
id	637,623	989,968	
salary	300,000	370,000	
doj	2013-11-01 00:00:00	2014-07-01 00:00:00	
10percentage	79	86	
12graduation	2,008	2,009	
12percentage	74	82	
collegeid	3,802	8,810	
collegetier	2	2	
collegegpa	72	76	
collegecityid	3,802	8,810	

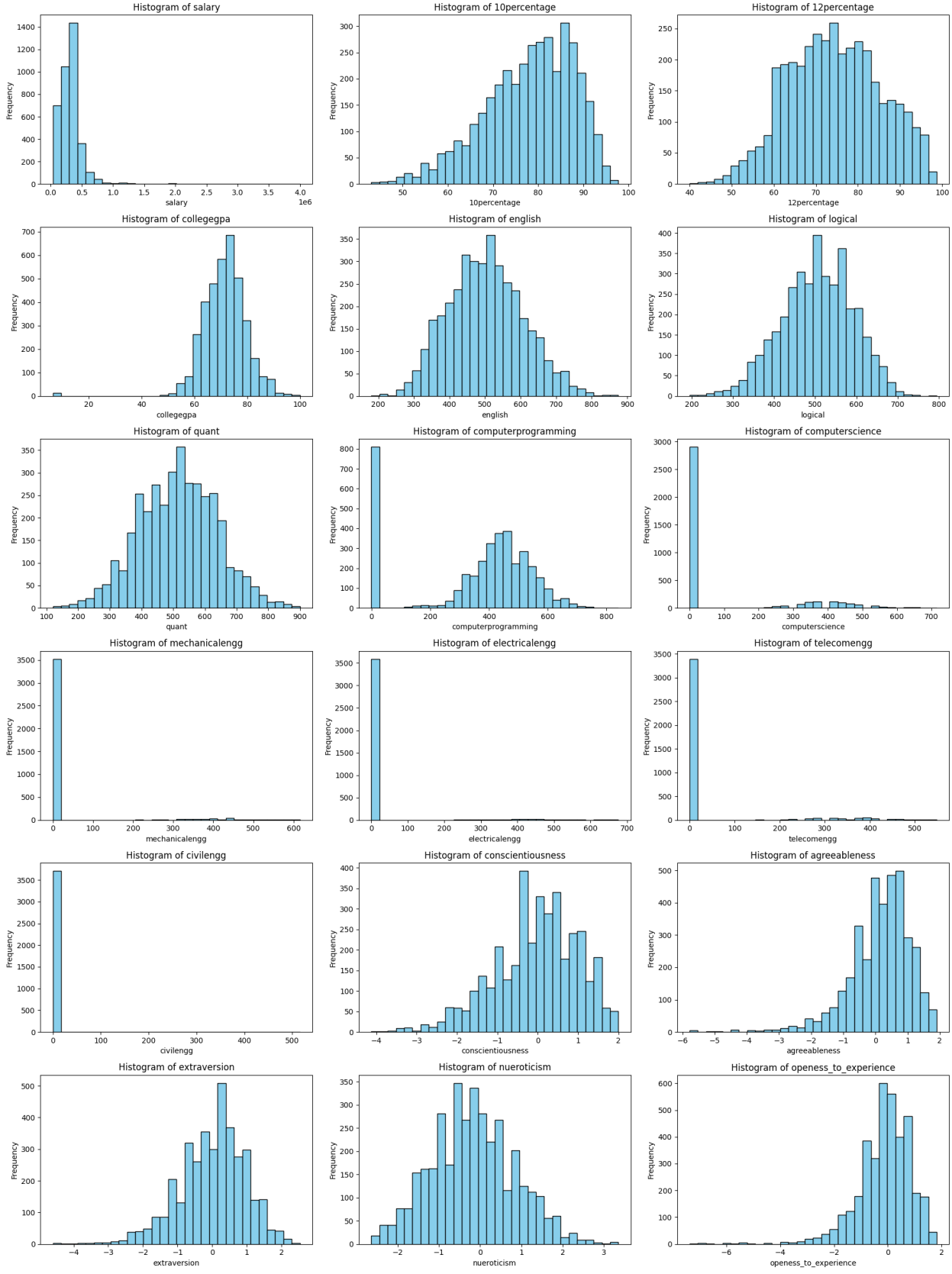
collegecitytier	0	1
graduationyear	2,013	2,014
english	500	570
logical	505	565
quant	515	605
domain	1	1
computerprogramming	415	495
electronicsandsemicon	-1	253
computerscience	0	0
mechanicalengg	0	0
electricalengg	0	0
telecomengg	0	0
civilengg	0	0
conscientiousness	0	1
agreeableness	0	1
extraversion	0	1
nueroticism	-0	1
openess_to_experience	-0	1

	max	std
id	1,298,275	363,317
salary	4,000,000	217,005
doj	2015-12-01 00:00:00	NaN
10percentage	98	10
12graduation	2,013	2
12percentage	99	11
collegeid	18,409	4,784
collegetier	2	0
collegegpa	100	8
collegecityid	18,409	4,784
collegecitytier	1	0
graduationyear	2,017	33
english	875	105
logical	795	87
quant	900	123
domain	1	0
computerprogramming	840	205
electronicsandsemicon	612	158
computerscience	715	175
mechanicalengg	616	98
electricalengg	676	87
telecomengg	548	104
civilengg	516	34
conscientiousness	2	1
agreeableness	2	1
extraversion	3	1
nueroticism	3	1
openess_to_experience	2	1

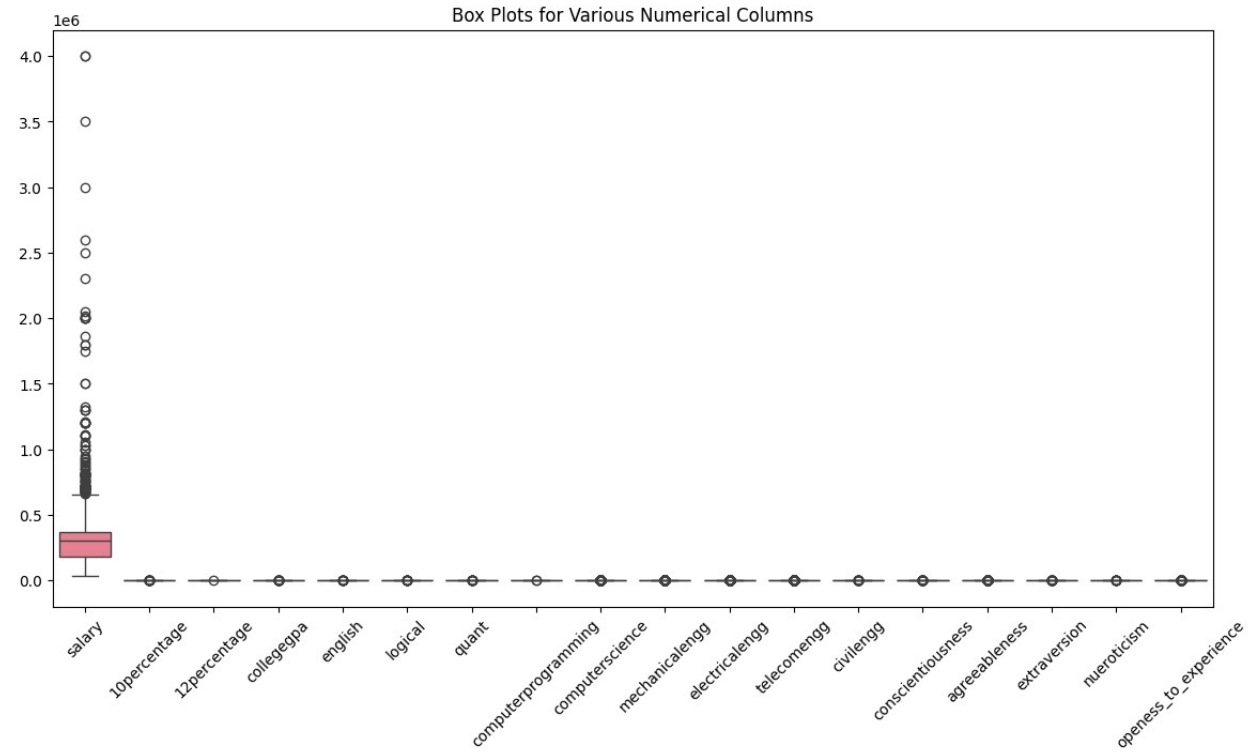
```
df.columns
```

```
Index(['id', 'salary', 'doj', 'dol', 'designation', 'jobcity',
'gender', 'dob',
       '10percentage', '10board', '12graduation', '12percentage',
'12board',
       'collegeid', 'collegetier', 'degree', 'specialization',
'collegegpa',
       'collegecityid', 'collegecitytier', 'collegestate',
'graduationyear',
       'english', 'logical', 'quant', 'domain', 'computerprogramming',
       'electronicsandsemicon', 'computerscience', 'mechanicalengg',
       'electricalengg', 'telecomengg', 'civilengg',
'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

```
columns_to_plot = ['salary', '10percentage', '12percentage',
'collegegpa', 'english', 'logical',
                   'quant', 'computerprogramming', 'computerscience',
'mechanicalengg',
                   'electricalengg', 'telecomengg', 'civilengg',
'conscientiousness',
                   'agreeableness', 'extraversion', 'nueroticism',
'openess_to_experience']

# Set up the figure and axes for subplots
fig, axes = plt.subplots(nrows=6, ncols=3, figsize=(18, 24))  # 6
rows, 3 columns layout
axes = axes.flatten()  # Flatten the 2D array of axes into 1D for
easier iteration

# Loop through each column and its respective axis
for i, column in enumerate(columns_to_plot):
    axes[i].hist(df[column].dropna(), bins=30, color='skyblue',
edgecolor='black')  # Plot histogram
    axes[i].set_title(f'Histogram of {column}')  # Set title for each
subplot
    axes[i].set_xlabel(column)  # X-axis label
    axes[i].set_ylabel('Frequency')  # Y-axis label

# Remove any unused subplots (if there are more axes than columns)
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

# Adjust layout to prevent overlapping
plt.tight_layout()

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

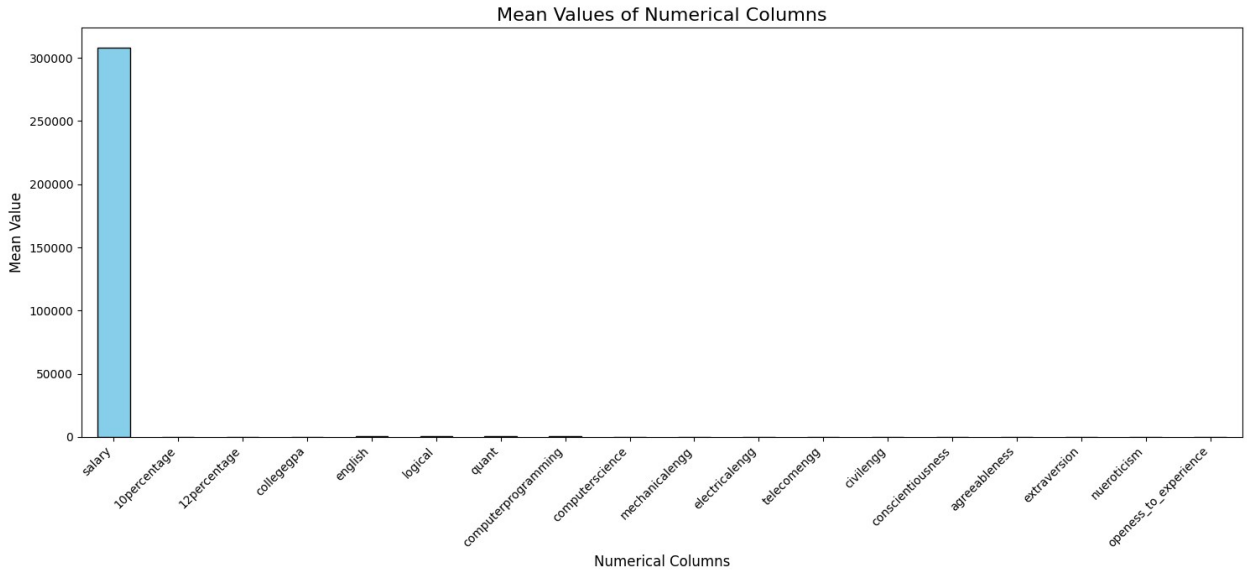

Histogram of salary
Frequency
salary
1e6
Histogram of 10percentage
10percentage
Histogram of 12percentage
12percentage
Histogram of collegegpa
collegegpa
Histogram of english
english
Histogram of logical
logical
Histogram of quant
quant
Histogram of computerprogramming
computerprogramming
Histogram of computerscience
computerscience
Histogram of mechanicalengg
mechanicalengg
Histogram of electricalengg
electricalengg
Histogram of telecomengg
telecomengg
Histogram of civilengg
civilengg
Histogram of conscientiousness
conscientiousness
Histogram of agreeableness
agreeableness
Histogram of extraversion
extraversion
Histogram of nueroticism
nueroticism
Histogram of openess_to_experience
openess_to_experience

```
# Correct list of columns to plot (only numerical columns)
columns_to_plot = ['salary', '10percentage', '12percentage',
'collegegpa',
                   'english', 'logical', 'quant',
'computerprogramming',
                   'computerscience', 'mechanicalengg',
'electricalengg',
                   'telecomengg', 'civilengg', 'conscientiousness',
                   'agreeableness', 'extraversion', 'nueroticism',
                   'openess_to_experience']

# Plot the box plot with valid columns
plt.figure(figsize=(14, 7))
sns.boxplot(data=df[columns_to_plot])
plt.title('Box Plots for Various Numerical Columns')
plt.xticks(rotation=45)
plt.show()
```



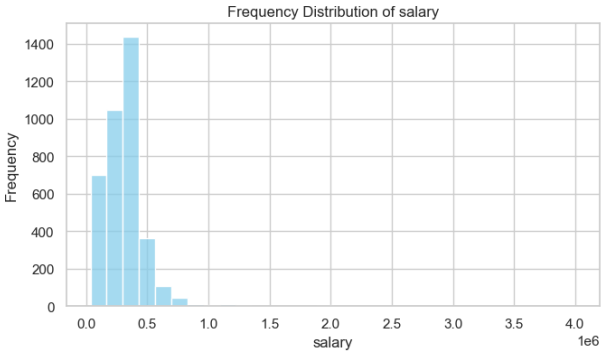

```
import matplotlib.pyplot as plt

# Select only numerical columns
columns_to_plot = ['salary', '10percentage', '12percentage',
'collegegpa', 'english', 'logical',
                   'quant', 'computerprogramming', 'computerscience',
'mechanicalengg',
                   'electricalengg', 'telecomengg', 'civilengg',
```

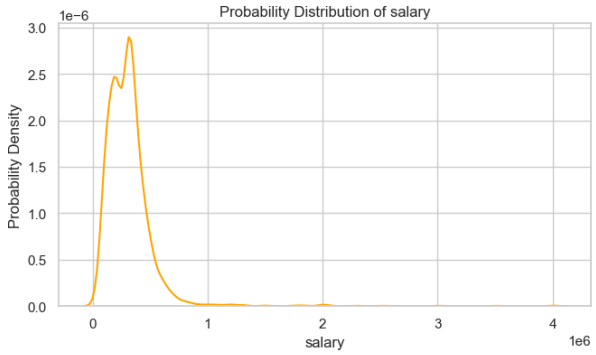
```
'conscientiousness',
                   'agreeableness', 'extraversion', 'nueroticism',
'openess_to_experience']

# Calculate the mean of each numerical column
mean_values = df[columns_to_plot].mean()

# Create the bar plot
plt.figure(figsize=(15, 7))  # Set the figure size
mean_values.plot(kind='bar', color='skyblue', edgecolor='black')

# Customize the plot
plt.title('Mean Values of Numerical Columns', fontsize=16)
plt.xlabel('Numerical Columns', fontsize=12)
plt.ylabel('Mean Value', fontsize=12)
plt.xticks(rotation=45, ha='right')  # Rotate x labels for better
visibility

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

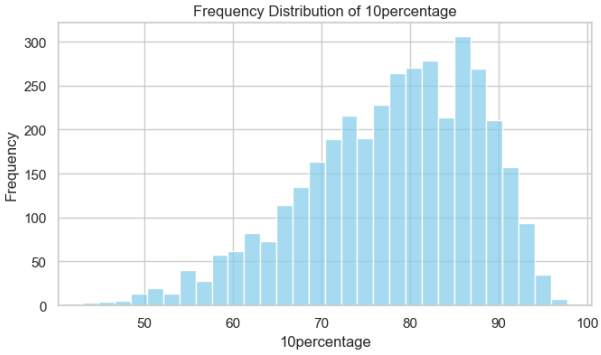


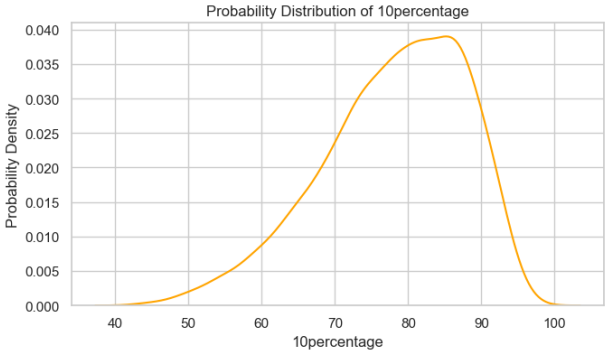

```
# Set the style of seaborn
sns.set(style="whitegrid")

# Define the columns for plotting
columns_to_plot = ['salary', '10percentage', '12percentage',
'collegegpa', 'english', 'logical',
                   'quant', 'computerprogramming', 'computerscience',
'mechanicalengg',
                   'electricalengg', 'telecomengg', 'civilengg',
'conscientiousness',
```

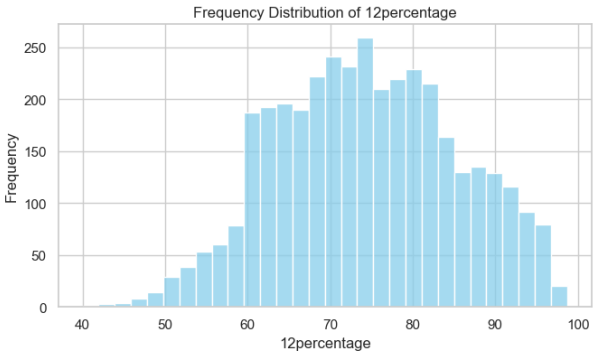
```
                    'agreeableness', 'extraversion', 'nueroticism',
'openess_to_experience']

# Create a figure with subplots
fig, axes = plt.subplots(nrows=len(columns_to_plot), ncols=2,
figsize=(14, len(columns_to_plot) * 4))

# Loop through each numerical column to plot
for i, column in enumerate(columns_to_plot):
    # Frequency Distribution
    sns.histplot(df[column], ax=axes[i, 0], bins=30, kde=False,
color='skyblue')
    axes[i, 0].set_title(f'Frequency Distribution of {column}',
fontsize=12)
    axes[i, 0].set_xlabel(column)
    axes[i, 0].set_ylabel('Frequency')

    # Probability Distribution (KDE)
    sns.kdeplot(df[column], ax=axes[i, 1], color='orange')
    axes[i, 1].set_title(f'Probability Distribution of {column}',
fontsize=12)
    axes[i, 1].set_xlabel(column)
    axes[i, 1].set_ylabel('Probability Density')

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

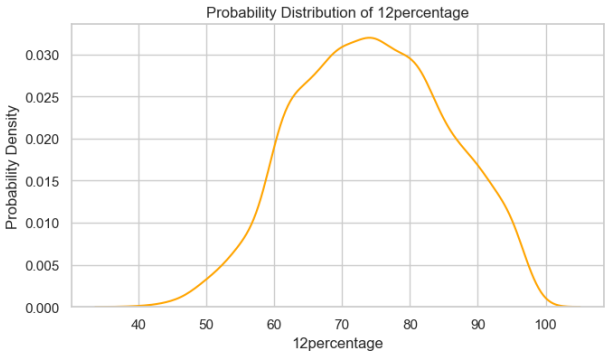

Frequency Distribution of salary
Frequency
salary
1e6

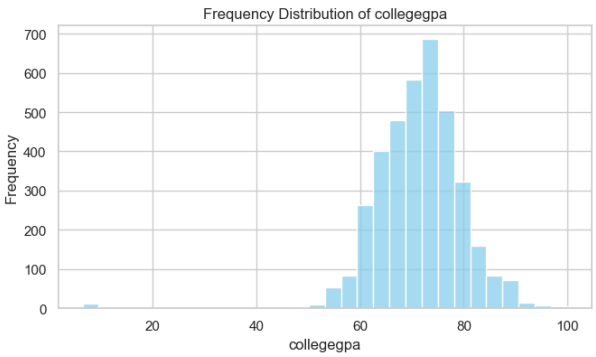

Probability Distribution of salary
1e−6
Probability Density
salary
1e6

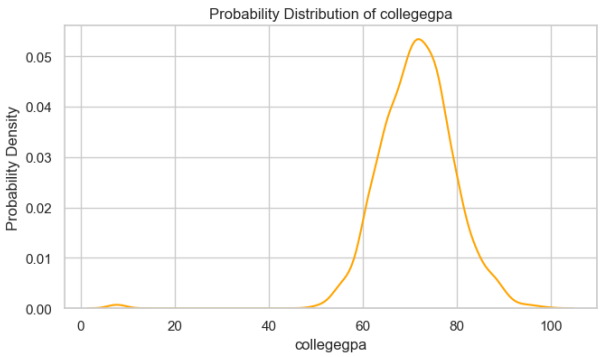

Frequency Distribution of 10percentage
Frequency
10percentage

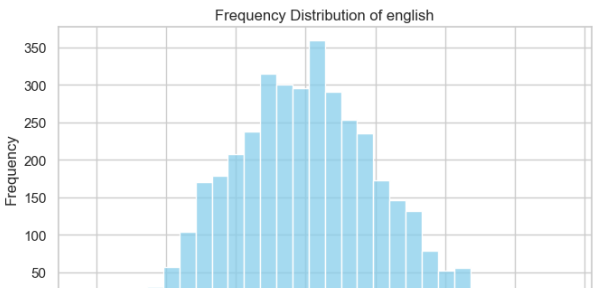

Probability Distribution of 10percentage
Probability Density
10percentage

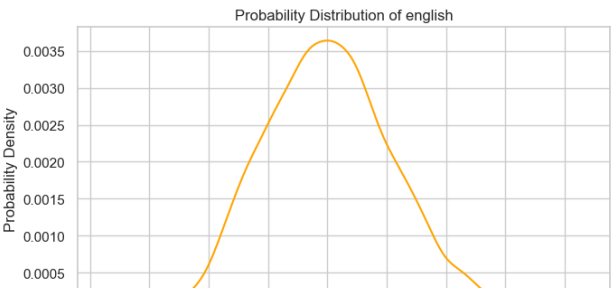

Frequency Distribution of 12percentage
Frequency
12percentage


Probability Distribution of 12percentage
Probability Density
12percentage


Frequency Distribution of collegegpa
Frequency
collegegpa

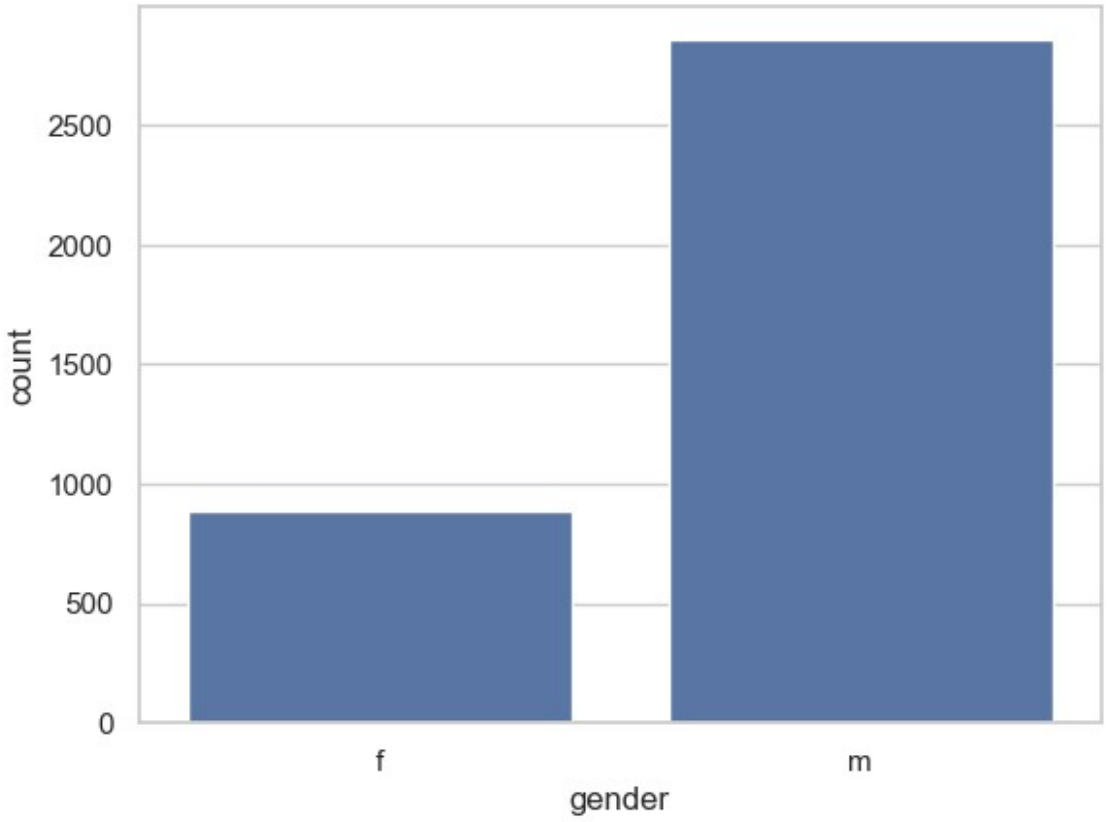

Probability Distribution of collegegpa
Probability Density
collegegpa


Frequency Distribution of english
Frequency

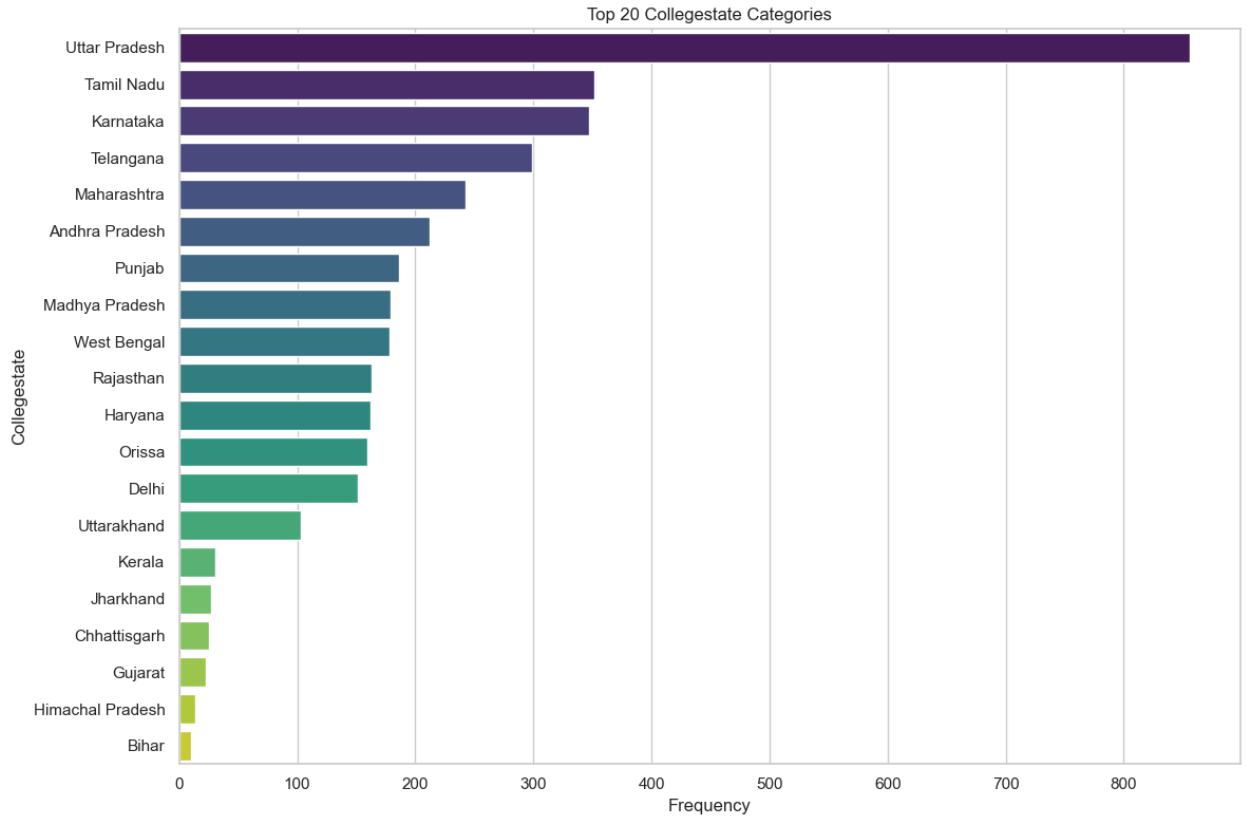

Probability Distribution of english
Probability Density

```
sns.countplot(x=df['gender'])
```

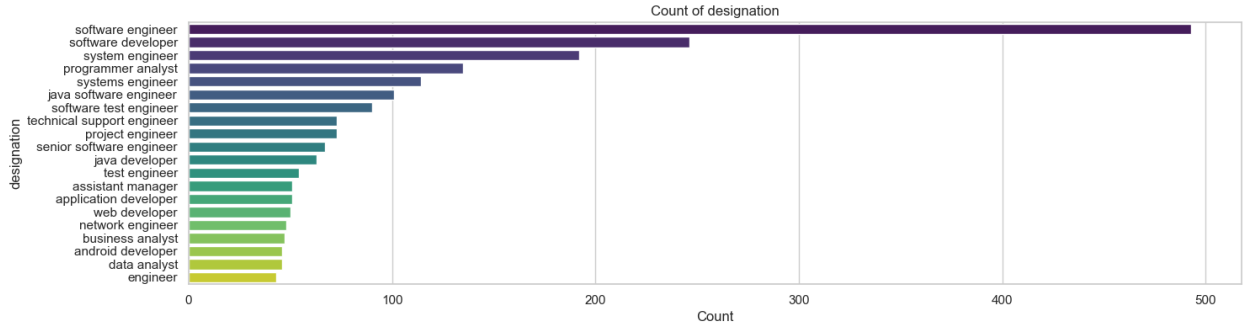
```
<Axes: xlabel='gender', ylabel='count'>
```

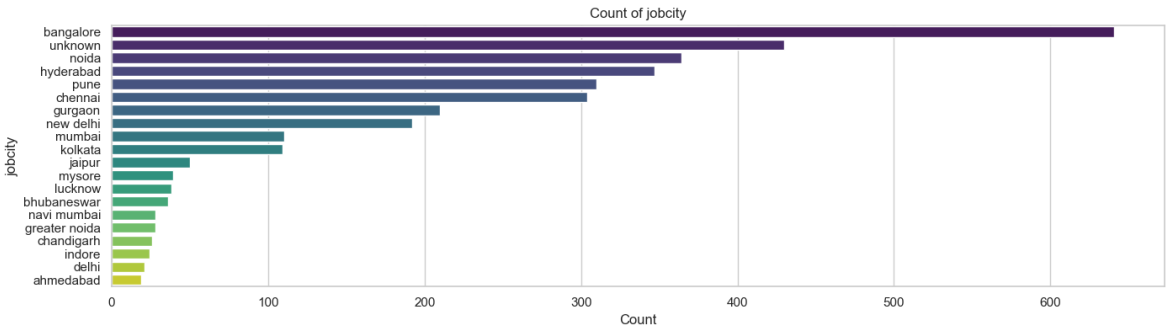


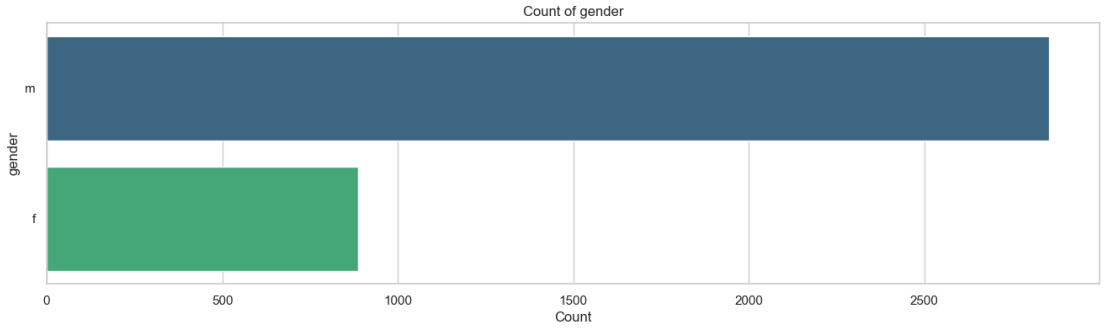

```
top_collegestates = df['collegestate'].value_counts().nlargest(20)
plt.figure(figsize=(12, 8))
sns.countplot(y='collegestate',
data=df[df['collegestate'].isin(top_collegestates.index)],
              palette='viridis', order=top_collegestates.index)
plt.title('Top 20 Collegestate Categories')
plt.xlabel('Frequency')
plt.ylabel('Collegestate')
plt.tight_layout()
plt.show()
```

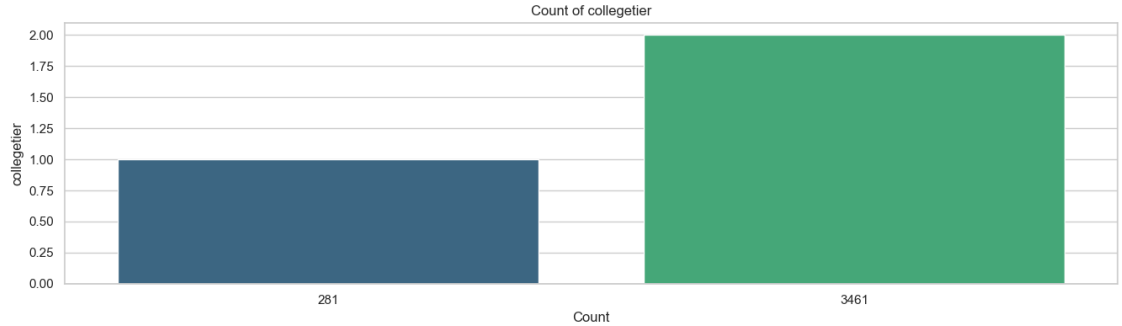


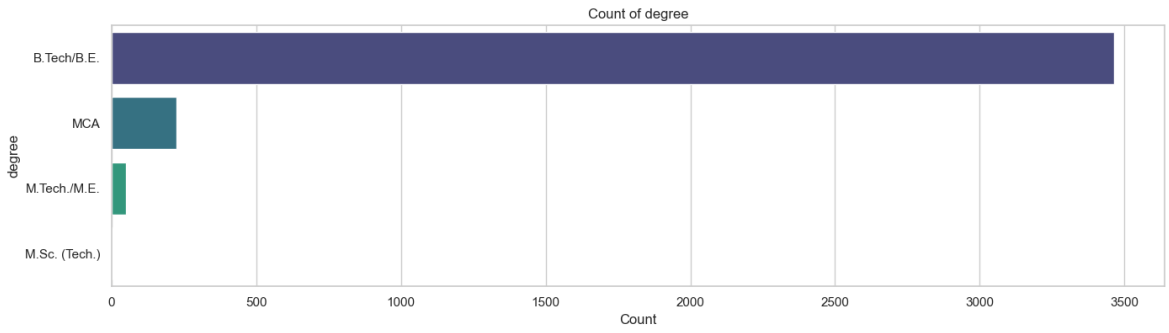

```
import matplotlib.pyplot as plt
import seaborn as sns

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

# List of important categorical columns
important_categorical_columns = ['designation', 'jobcity', 'gender',
'collegetier', 'degree']

# Create a bar plot for each important categorical column
plt.figure(figsize=(15, 20))  # Adjust the figure size as needed
for i, column in enumerate(important_categorical_columns):
    plt.subplot(len(important_categorical_columns), 1, i + 1)  #
Create a subplot for each column
    top_values = df[column].value_counts().nlargest(20)  # Get top 20
values
    sns.barplot(x=top_values.values, y=top_values.index,
palette='viridis')  # Horizontal bar plot
    plt.title(f'Count of {column}')  # Set the title
    plt.xlabel('Count')  # Label for x-axis
    plt.ylabel(column)  # Label for y-axis

plt.tight_layout()  # Adjust layout to prevent clipping of tick-labels
plt.show()
```


Count of designation
software engineer
software developer
system engineer
programmer analyst
systems engineer
java software engineer
software test engineer
technical support engineer
project engineer
senior software engineer
java developer
test engineer
assistant manager
application developer
web developer
network engineer
business analyst
android developer
data analyst
engineer
designation
0
100
200
300
400
500
Count


Count of jobcity
bangalore
unknown
noida
hyderabad
pune
chennai
gurgaon
new delhi
mumbai
kolkata
jaipur
mysore
lucknow
bhubaneswar
navi mumbai
greater noida
chandigarh
indore
delhi
ahmedabad
jobcity
0
100
200
300
400
500
600
Count


Count of gender
m
f
gender
0
500
1000
1500
2000
2500
Count


Count of collegetier
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
collegetier
281
3461
Count


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

```
df.columns
```

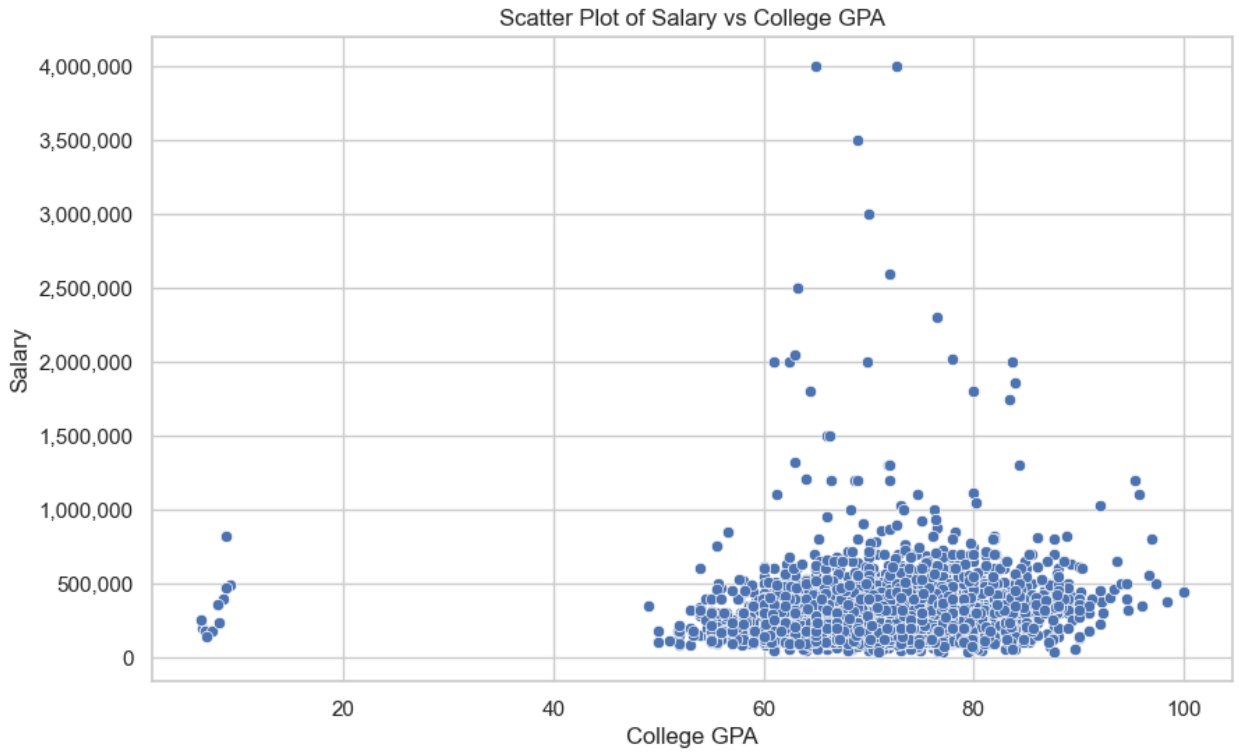
```
Index(['id', 'salary', 'doj', 'dol', 'designation', 'jobcity',
'gender', 'dob',
       '10percentage', '10board', '12graduation', '12percentage',
'12board',
       'collegeid', 'collegetier', 'degree', 'specialization',
'collegegpa',
       'collegecityid', 'collegecitytier', 'collegestate',
'graduationyear',
       'english', 'logical', 'quant', 'domain', 'computerprogramming',
       'electronicsandsemicon', 'computerscience', 'mechanicalengg',
       'electricalengg', 'telecomengg', 'civilengg',
'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

```
df.shape
```

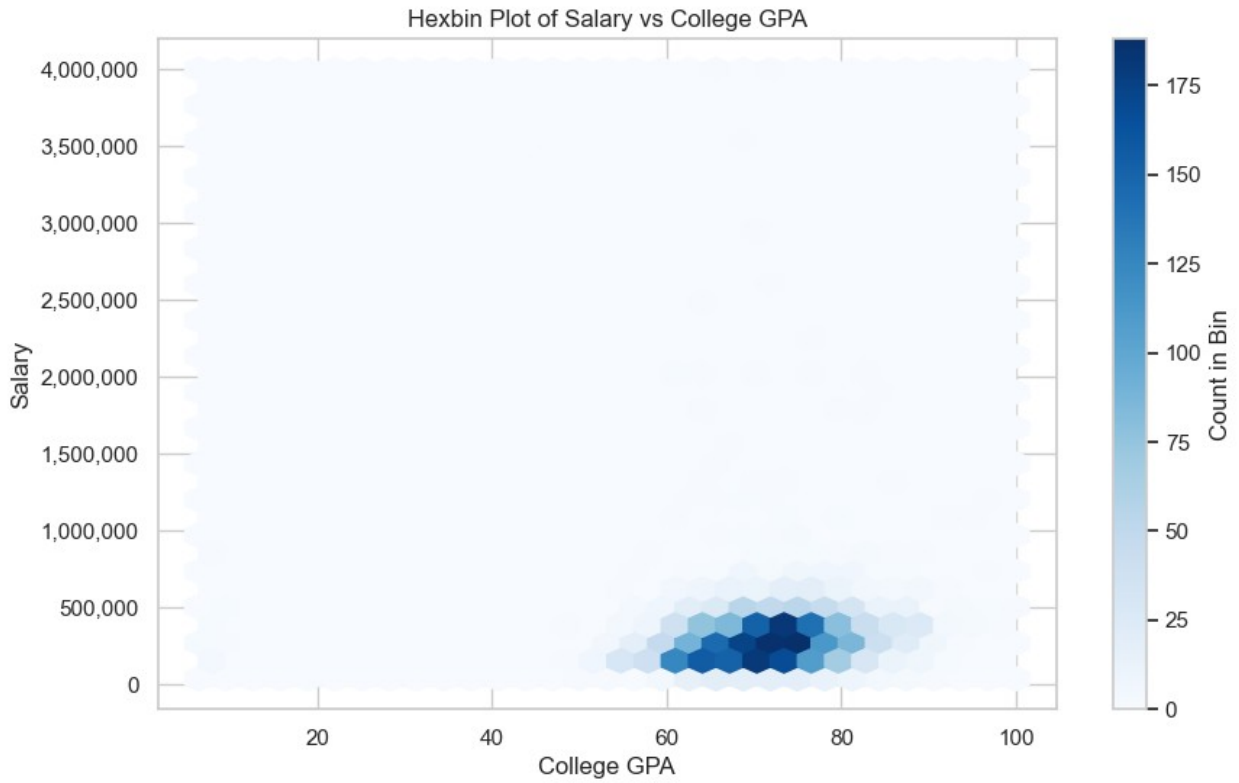
```
(3743, 38)
```

Bivariate Analysis

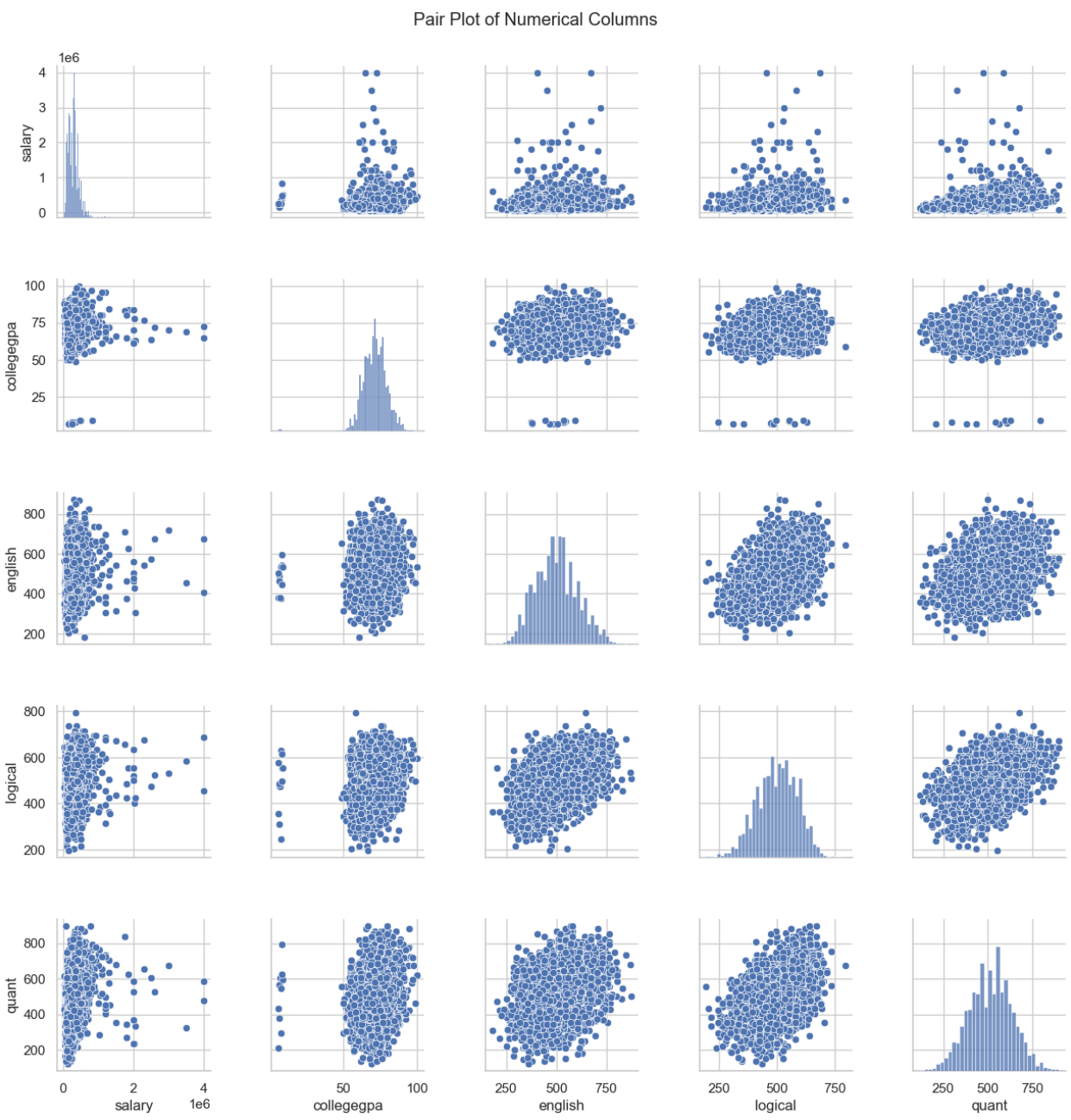
```
from matplotlib.ticker import FuncFormatter

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

plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='collegegpa', y='salary')
plt.title('Scatter Plot of Salary vs College GPA')
plt.xlabel('College GPA')
plt.ylabel('Salary')
plt.grid(True)

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

plt.show()
```



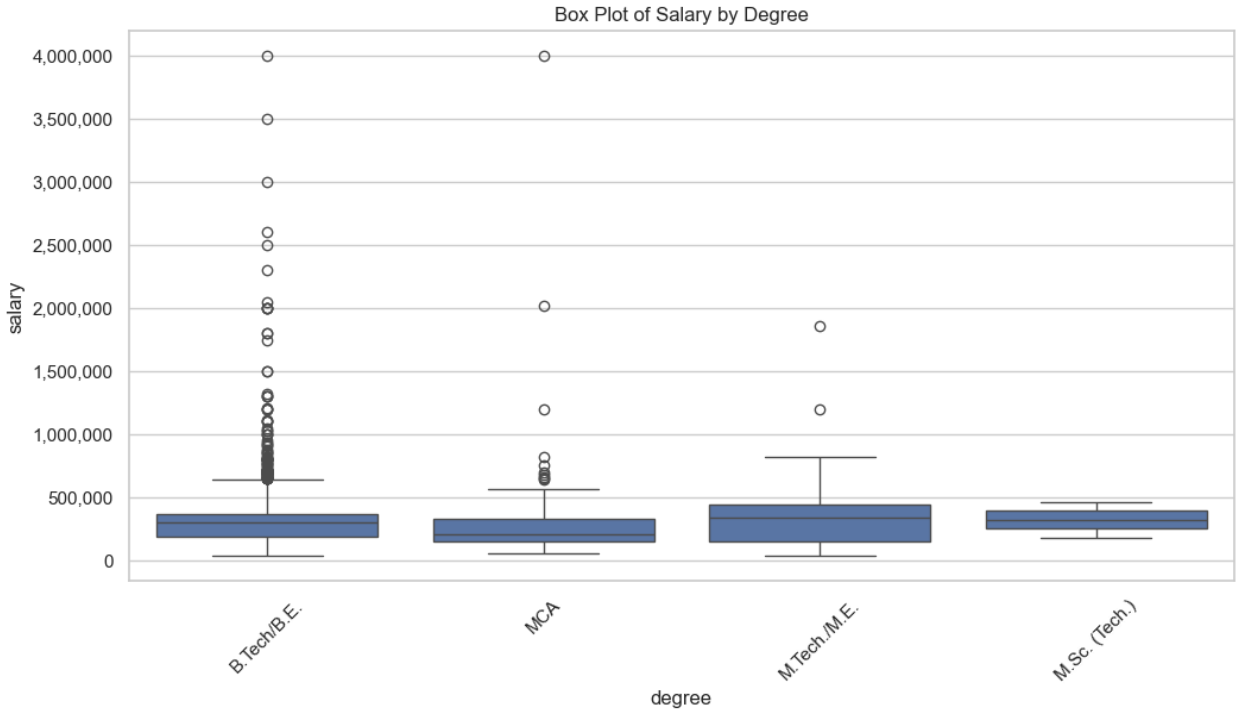

```
plt.figure(figsize=(10, 6))
plt.hexbin(df['collegegpa'], df['salary'], gridsize=30, cmap='Blues')
plt.colorbar(label='Count in Bin')
plt.title('Hexbin Plot of Salary vs College GPA')
plt.xlabel('College GPA')
plt.ylabel('Salary')
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))
plt.show()
```



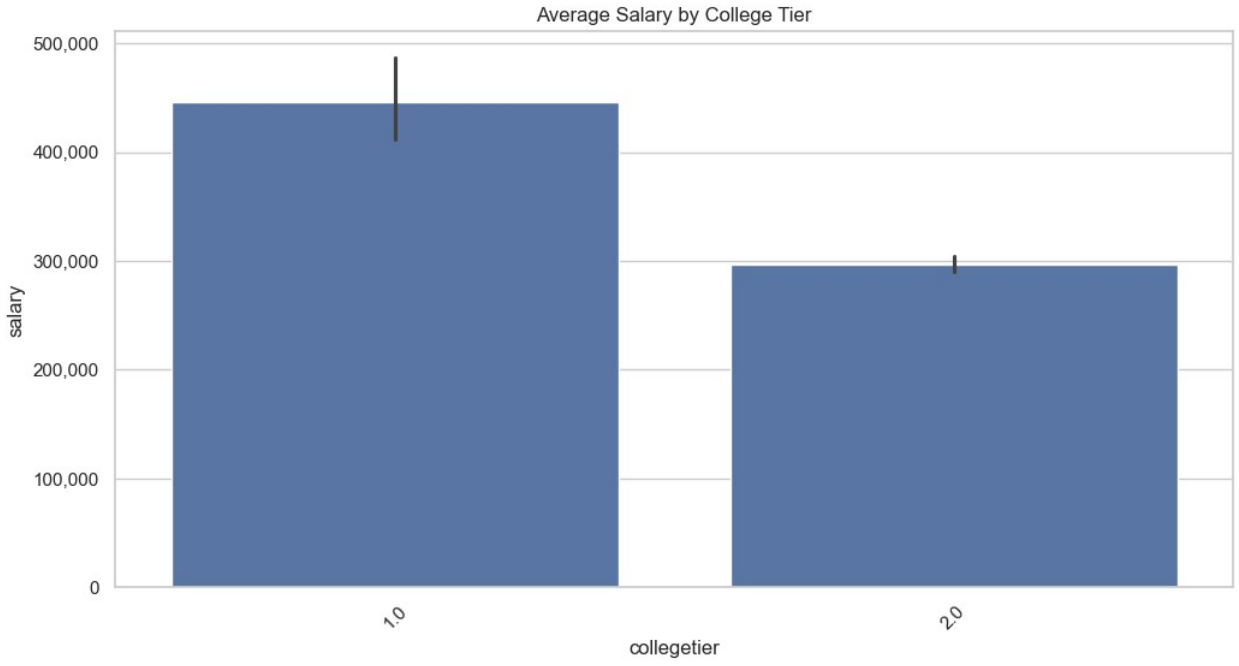

```
numerical_columns = ['salary', 'collegegpa', 'english', 'logical',
'quant']
sns.set(style="whitegrid")
pair_plot = sns.pairplot(df[numerical_columns])
plt.suptitle('Pair Plot of Numerical Columns', y=1.02)
plt.subplots_adjust(hspace=0.4, wspace=0.4)
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))

plt.show()
```



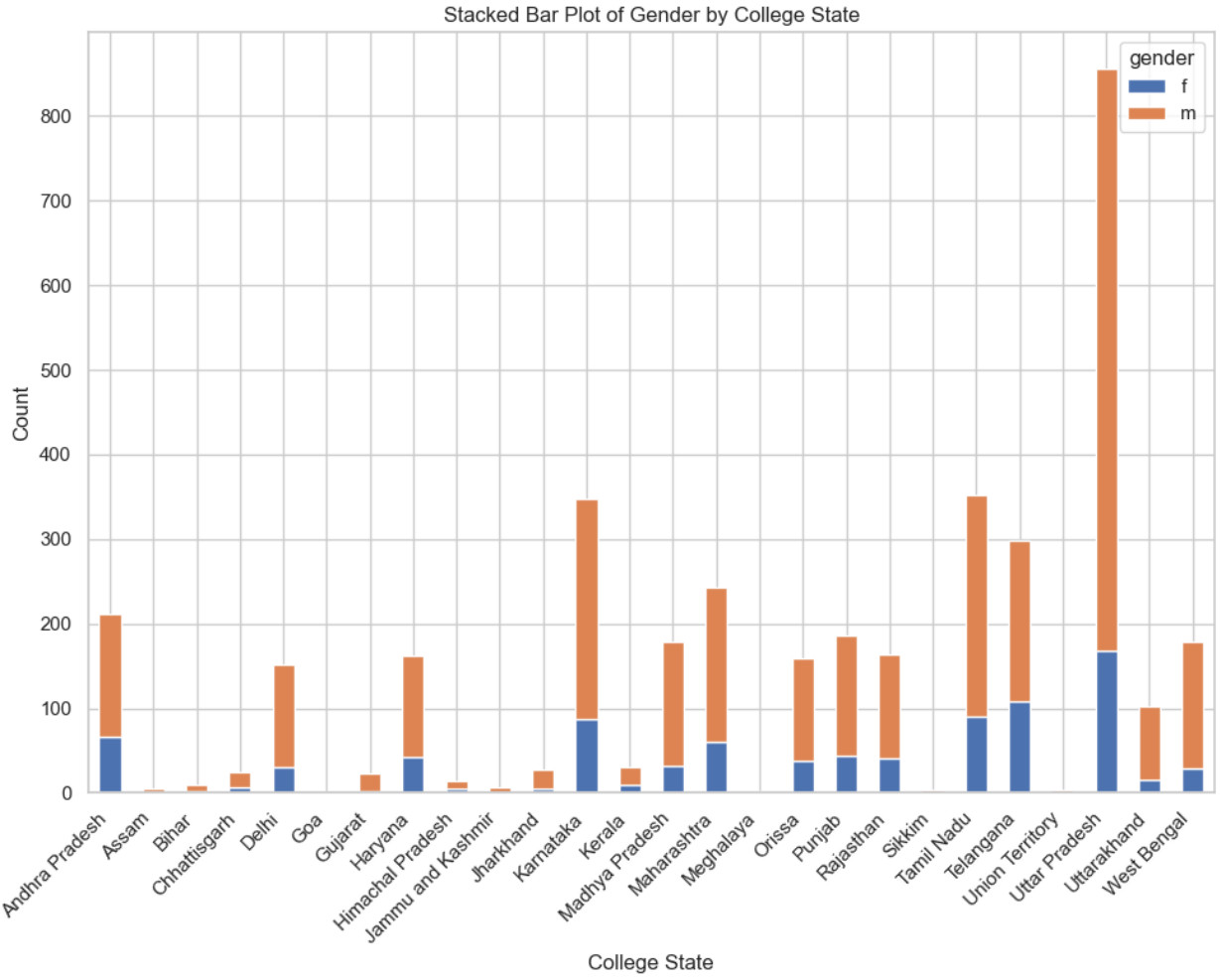

```
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='degree', y='salary')
plt.title('Box Plot of Salary by Degree')
plt.xticks(rotation=45)
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))
plt.show()
```




```
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='collegetier', y='salary', estimator=np.mean)
plt.title('Average Salary by College Tier')
plt.xticks(rotation=45)
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))
plt.show()
```



```
pivot_table = df.pivot_table(index='collegestate', columns='gender',
values='salary', aggfunc='count').fillna(0)
pivot_table.plot(kind='bar', stacked=True, figsize=(10, 8))
plt.title('Stacked Bar Plot of Gender by College State')
plt.xlabel('College State')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')  # Adjusted alignment to 'right'
plt.tight_layout()  # Adjust layout to prevent clipping
plt.show()
```



Research Questions

```
df.columns

Index(['id', 'salary', 'doj', 'dol', 'designation', 'jobcity',
'gender', 'dob',
       '10percentage', '10board', '12graduation', '12percentage',
'12board',
       'collegeid', 'collegetier', 'degree', 'specialization',
```

```
'collegegpa',
       'collegecityid', 'collegecitytier', 'collegestate',
'graduationyear',
       'english', 'logical', 'quant', 'domain', 'computerprogramming',
       'electronicsandsemicon', 'computerscience', 'mechanicalengg',
       'electricalengg', 'telecomengg', 'civilengg',
'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

```
from scipy import stats

# Specify the claimed salary range
lower_bound = 1 * 100000  # converting lakhs to actual number
upper_bound = 5 * 100000

# Filter data for specified job titles
job_titles = ['Programming Analyst', 'Software Engineer', 'Hardware
Engineer', 'Associate Engineer']
filtered_data = df[df['designation'].isin(job_titles)]

# Perform one-sample t-test on salary
if not filtered_data.empty:
    t_statistic, p_value = stats.ttest_1samp(filtered_data['salary'],
lower_bound)

    # Display the results
    print(f"T-statistic: {t_statistic}, P-value: {p_value}")

    # Interpret the p-value
    alpha = 0.05
    if p_value < alpha:
        print("Reject the null hypothesis: Average salary
significantly differs from the claimed range.")
    else:
        print("Fail to reject the null hypothesis: Average salary does
not significantly differ from the claimed range.")
else:
    print("No data found for the specified job titles.")
```

```
No data found for the specified job titles.
```

```
# Assuming df is your DataFrame containing the data
job_titles = ['Programming Analyst', 'Software Engineer', 'Hardware
Engineer', 'Associate Engineer']
salary_data = df[df['designation'].isin(job_titles)]

# Calculate the average salary for each job title
average_salaries = salary_data.groupby('designation')
['salary'].mean().reset_index()
```

```python
# Check if average salaries are within the claimed range of 2.5 to 3 Lakhs
average_salaries['within_claimed_range'] = average_salaries['salary'].apply(lambda x: 2.5 <= x <= 3)

print("Average Salaries for Specified Job Titles:")
print(average_salaries)

print("\nAverage Salaries within Claimed Range:")
print(average_salaries[average_salaries['within_claimed_range']])
```

```
Average Salaries for Specified Job Titles:
Empty DataFrame
Columns: [designation, salary, within_claimed_range]
Index: []

Average Salaries within Claimed Range:
Empty DataFrame
Columns: []
Index: []
```

```python
# Create a contingency table
contingency_table = pd.crosstab(df['gender'], df['specialization'])

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

# Perform Chi-Square test
chi2_stat, p_value, dof, expected = stats.chi2_contingency(contingency_table)

# Create a results DataFrame with reset index
results = pd.DataFrame({
    'Metric': ['Chi-Squared Statistic', 'P-value', 'Degrees of Freedom', 'Conclusion'],
    'Value': [
        chi2_stat,
        p_value,
        dof,
        "Reject the null hypothesis" if p_value < 0.05 else "Fail to reject the null hypothesis"
    ]
})

# Reset the index of the results DataFrame
results.reset_index(drop=True, inplace=True)

# Display the results
```

```
print("\nChi-Square Test Results:")
print(results)
```

```
Contingency Table:
specialization  aeronautical engineering  \
gender
f                                       1
m                                       2

specialization  applied electronics and instrumentation  \
gender
f                                                      1
m                                                      7

specialization  automobile/automotive engineering  biomedical
engineering  \
gender

f                                                  0
2
m                                                  4
0

specialization  biotechnology  ceramic engineering  chemical
engineering  \
gender

f                            9                     0
1
m                            6                     1
7

specialization  civil engineering  computer and communication
engineering  \
gender

f                                6
0
m                               17
1

specialization  computer application  ...  internal combustion engine
\
gender                                  ...

f                                  55  ...                            0

m                                 171  ...                            1


specialization  mechanical & production engineering  \
```

```
gender
f                                                    0
m                                                    1

specialization  mechanical and automation  mechanical engineering  \
gender
f                                          0                       9
m                                          4                     182

specialization  mechatronics  metallurgical engineering  other  \
gender
f                           1                          0      0
m                           2                          2     12

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:
                  Metric                      Value
0  Chi-Squared Statistic                        101
1                P-value                          0
2    Degrees of Freedom                         45
3             Conclusion  Reject the null hypothesis
```