```
 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(r"C:\Users\maazp\Downloads\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
...         ...     ...        ...           ...           ...
3993      train   47916   280000.0  10/1/11 0:00  10/1/12 0:00
3994      train  752781   100000.0   7/1/13 0:00   7/1/13 0:00
3995      train  355888   320000.0   7/1/13 0:00       present
3996      train  947111   200000.0   7/1/14 0:00   1/1/15 0:00
3997      train  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  Asifabadbanglore      f  3/20/92
0:00
3997     senior systems engineer           Chennai      f  2/26/91
0:00
```

```
      10percentage  ... ComputerScience  MechanicalEngg
ElectricalEngg  \
0            84.30  ...              -1              -1               -
1
1            85.40  ...              -1              -1               -
1
2            85.00  ...              -1              -1               -
1
3            85.60  ...              -1              -1               -
1
4            78.00  ...              -1              -1               -
1
...            ...  ...             ...             ...              ..
.
3993         52.09  ...              -1              -1               -
1
3994         90.00  ...              -1              -1               -
1
3995         81.86  ...              -1              -1               -
1
3996         78.72  ...             438              -1               -
1
3997         70.60  ...              -1              -1               -
1

     TelecomEngg  CivilEngg  conscientiousness agreeableness
extraversion  \
0             -1         -1             0.9737        0.8128
0.5269
1             -1         -1            -0.7335        0.3789
1.2396
2             -1         -1             0.2718        1.7109
0.1637
3             -1         -1             0.0464        0.3448      -
0.3440
4             -1         -1            -0.8810       -0.2793      -
1.0697
...          ...        ...                ...           ...
...
3993          -1         -1            -0.1082        0.3448
0.2366
3994          -1         -1            -0.3027        0.8784
0.9322
3995          -1         -1            -1.5765       -1.5273      -
1.5051
3996          -1         -1            -0.1590        0.0459      -
0.4511
3997          -1         -1            -1.1128       -0.2793      -
```

```
0.6343

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

[3998 rows x 39 columns]

 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             3998 non-null   object
 1   ID                     3998 non-null   int64
 2   Salary                 3998 non-null   float64
 3   DOJ                    3998 non-null   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
```

```
if 'Unnamed: 0' in df.columns:
    df = df.drop('Unnamed: 0', axis=1)
else:
    print("Column 'Unnamed: 0' not found in DataFrame.")
```

```
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              -
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]
```

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

```
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
```

```
 df.shape
```

```
(3998, 38)
```

1 Data Cleaning

```
 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', '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)
```

```
 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 )' 'mainpuri' 'dammam' 'haldia' 'rae bareli'
 'patiala' 'gorakhpur' 'new dehli' 'ambala city' 'karad' 'rajpura'
 'haryana' 'asifabadbanglore']

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',
}


 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
...      ...         ...         ...           ...
...
3993   47916   280000.0 2011-10-01  10/1/12 0:00            software
engineer
3994  752781   100000.0 2013-07-01   7/1/13 0:00             technical
writer
3995  355888   320000.0 2013-07-01       present  associate software
engineer
3996  947111   200000.0 2014-07-01   1/1/15 0:00           software
developer
3997  324966   400000.0 2013-02-01       present      senior systems
engineer

        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
...         ...    ...           ...           ...
3993  new delhi      m  4/15/87 0:00         52.09
3994  hyderabad      f  8/27/92 0:00         90.00
3995  bangalore      m   7/3/91 0:00         81.86
3996  bangalore      f  3/20/92 0:00         78.72
3997    chennai      f  2/26/91 0:00         70.60

                              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
...                               ...  ...              ...
...
3993                             cbse  ...               -1
```

```
-1
3994                     state board  ...                -1
-1
3995                      bse,odisha  ...                -1
-1
3996                     state board  ...               438
-1
3997                             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
...             ...          ...        ...               ...
...
3993             -1           -1         -1           -0.1082
0.3448
3994             -1           -1         -1           -0.3027
0.8784
3995             -1           -1         -1           -1.5765       -
1.5273
3996             -1           -1         -1           -0.1590
0.0459
3997             -1           -1         -1           -1.1128       -
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
...            ...          ...                    ...
3993        0.2366      0.64980                -0.9194
3994        0.9322      0.77980                -0.0943
3995       -1.5051     -1.31840                -0.7615
3996       -0.4511     -0.36120                -0.0943
3997       -0.6343      1.32553                -0.6035

[3998 rows x 38 columns]
```

```
 # Replace date values with "Left" in dol
df['dol'] = df['dol'].apply(lambda x: "Left" if x != "present" else x)

 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               -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]
```

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

```
dol
Left       2123
present    1875
Name: count, dtype: int64
```

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

```
307699.85
```

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

```
4000000.0
```

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

```
35000.0
```

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

```
gender
m    3041
f     957
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
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]
```

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

```
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
```

```
pd.options.display.float_format = '{:,.0f}'.format
# Display the describe() output for the 'salary' column
df.describe().transpose()
```

```
                      count                           mean  \
id                    3,998                        663,795
salary                3,998                        307,700
doj                    3998  2013-07-02 11:04:10.325162496
10percentage          3,998                             78
12graduation          3,998                          2,008
12percentage          3,998                             74
collegeid             3,998                          5,157
collegetier           3,998                              2
collegegpa            3,998                             71
collegecityid         3,998                          5,157
collegecitytier       3,998                              0
graduationyear        3,998                          2,012
english               3,998                            502
logical               3,998                            502
quant                 3,998                            513
domain                3,998                              1
computerprogramming   3,998                            353
electronicsandsemicon 3,998                             95
computerscience       3,998                             92
mechanicalengg        3,998                             24
electricalengg        3,998                             17
telecomengg           3,998                             33
civilengg             3,998                              4
conscientiousness     3,998                             -0
agreeableness         3,998                              0
extraversion          3,998                              0
nueroticism           3,998                             -0
openess_to_experience 3,998                             -0

                                       min                  25%  \
id                                  11,244              334,284
salary                              35,000              180,000
doj                   1991-06-01 00:00:00  2012-10-01 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	639,600	990,480	
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	83	
collegeid	3,879	8,818	
collegetier	2	2	
collegegpa	72	76	
collegecityid	3,879	8,818	
collegecitytier	0	1	
graduationyear	2,013	2,014	
english	500	570	
logical	505	565	
quant	515	595	
domain	1	1	
computerprogramming	415	495	
electronicsandsemicon	-1	233	
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
```

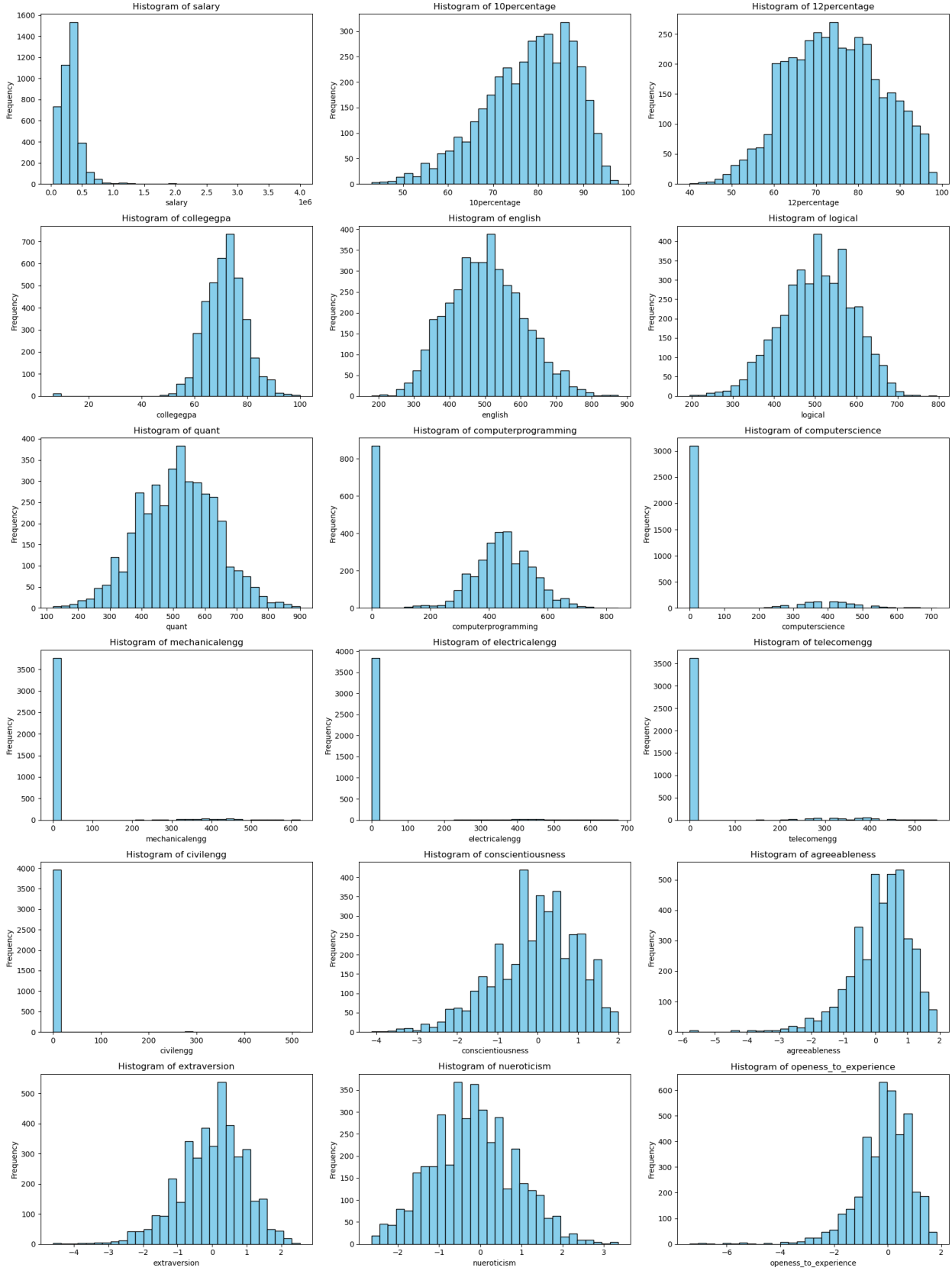
```
                                       max     std
id                               1,298,275 363,218
salary                           4,000,000 212,737
doj                  2015-12-01 00:00:00     NaN
10percentage                          98      10
12graduation                       2,013       2
12percentage                          99      11
collegeid                         18,409   4,802
collegetier                            2       0
collegegpa                           100       8
collegecityid                     18,409   4,802
collegecitytier                        1       0
graduationyear                     2,017      32
english                              875     105
logical                              795      87
quant                                900     122
domain                                 1       0
computerprogramming                  840     205
electronicsandsemicon                612     158
computerscience                      715     175
mechanicalengg                       623      98
electricalengg                       676      87
telecomengg                          548     105
civilengg                            516      37
conscientiousness                      2       1
agreeableness                          2       1
extraversion                           3       1
nueroticism                            3       1
openess_to_experience                  2       1
```

```
df.columns
```

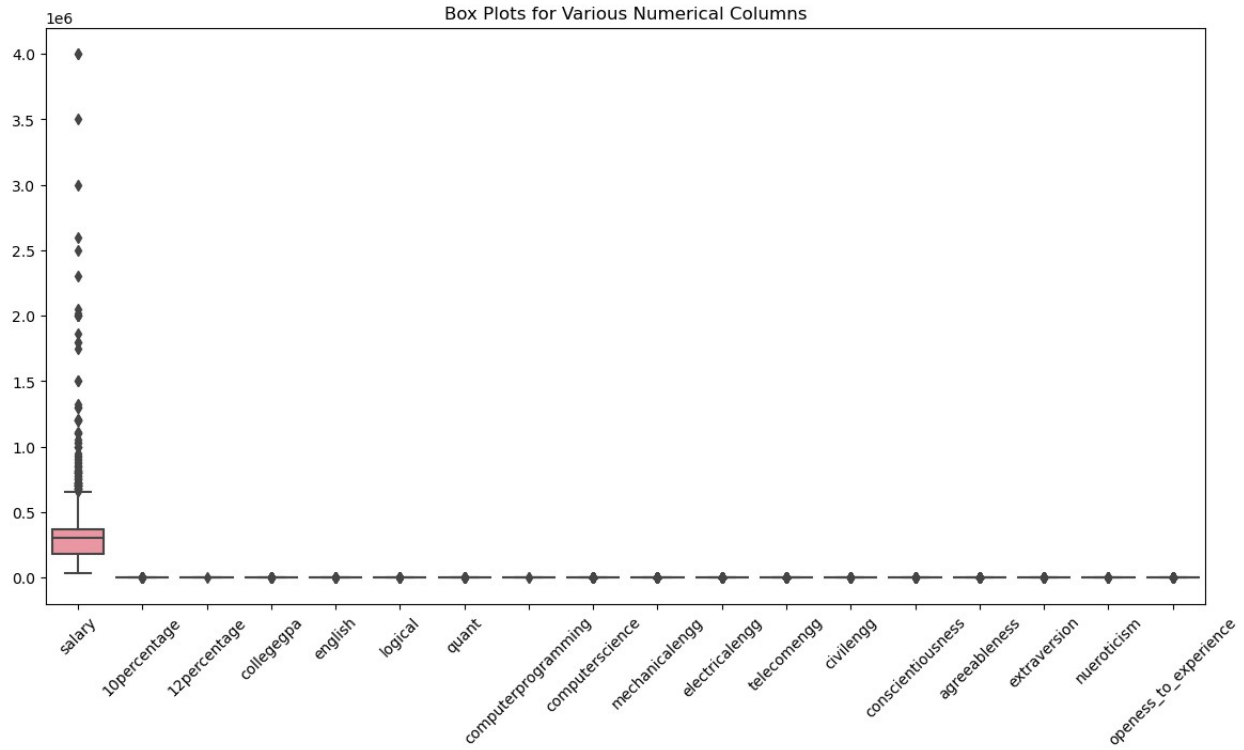
```
Index(['id', 'salary', 'doj', 'dol', 'designation', 'jobcity',
'gender', 'dob',
       '10percentage', '10board', '12graduation', '12percentage',
'12board',
       'collegeid', 'collegetier', 'degree', 'specialization',
'collegegpa',
       'collegecityid', 'collegecitytier', 'collegestate',
'graduationyear',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

```
       'openess_to_experience'],
      dtype='object')
```

```
import matplotlib.pyplot as plt

# Select the columns you want to plot
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()
```

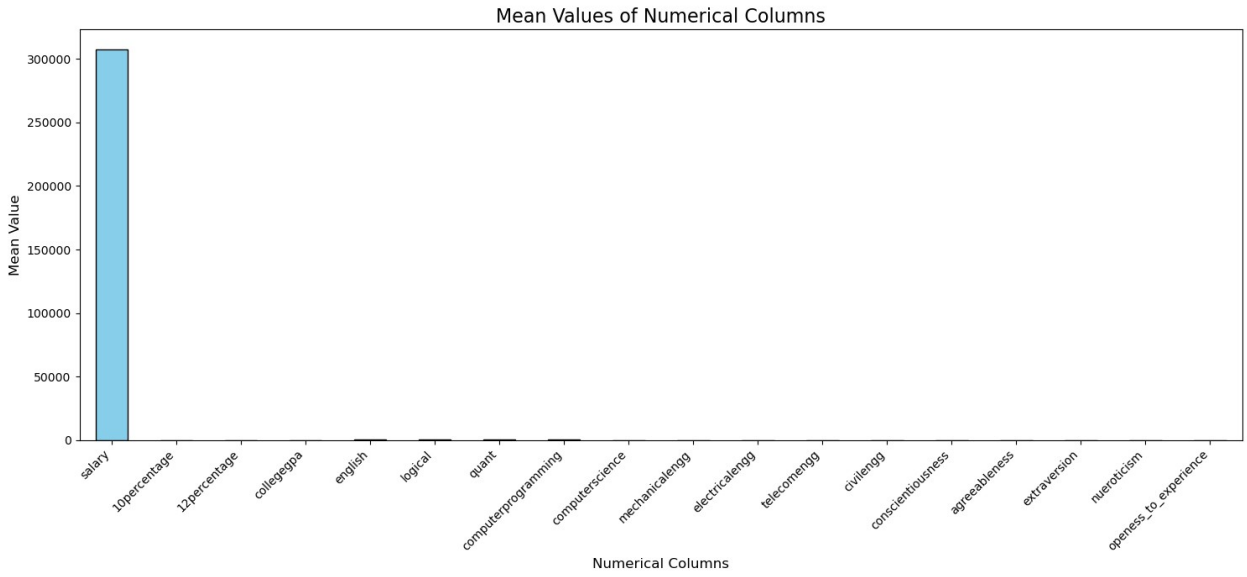

Histogram of salary
Frequency
salary
1e6
Histogram of 10percentage
10percentage
Histogram of 12percentage
12percentage
Histogram of collegegpa
collegegpa
Histogram of english
english
Histogram of logical
logical
Histogram of quant
quant
Histogram of computerprogramming
computerprogramming
Histogram of 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()
```



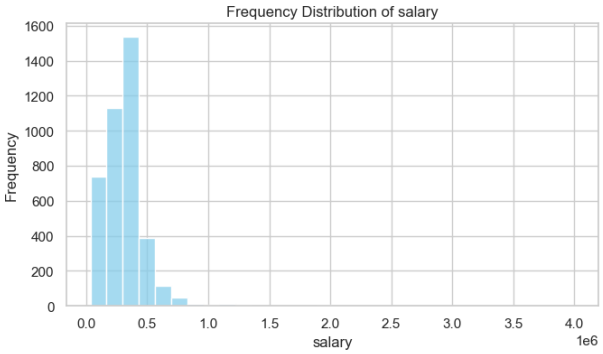

```
import matplotlib.pyplot as plt

# 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'
]
```

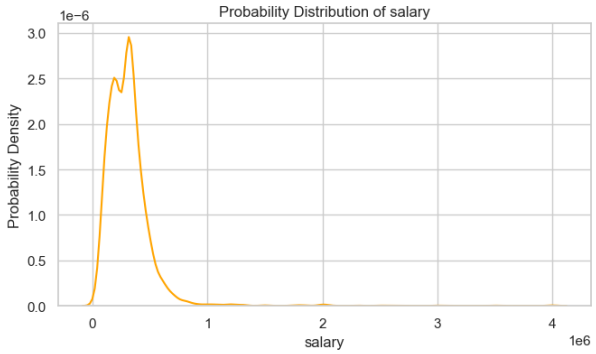
```
# Calculate the mean of each numerical column
mean_values = df[columns_to_plot].mean()

# 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()
```

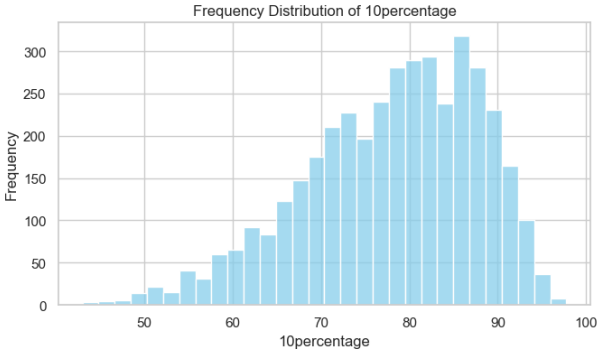


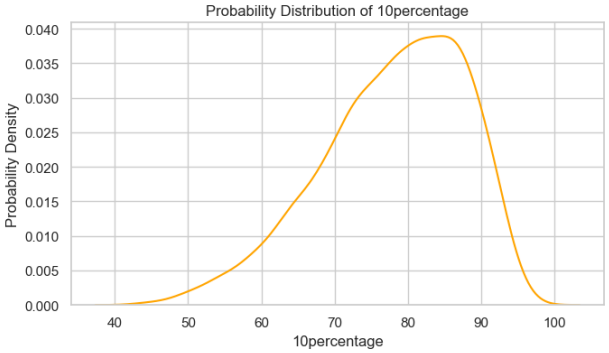

```
import seaborn as sns
import matplotlib.pyplot as plt

# 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',
```

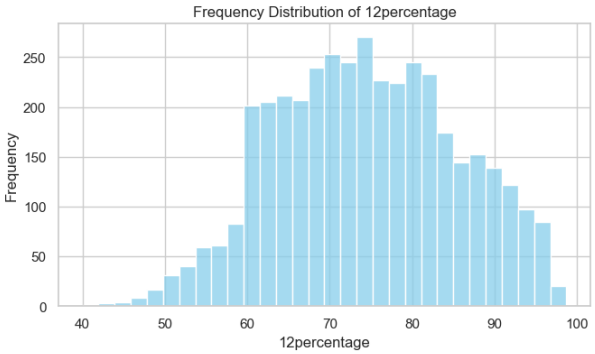
```
    'telecomengg', 'civilengg', 'conscientiousness',
    'agreeableness', 'extraversion', 'nueroticism',
    'openess_to_experience'
]

# 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()
```

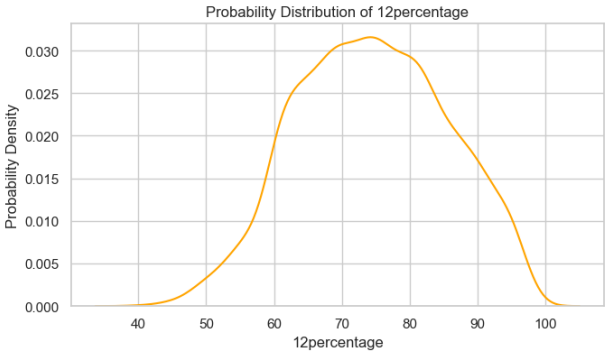

Frequency Distribution of salary
Frequency
salary
1e6

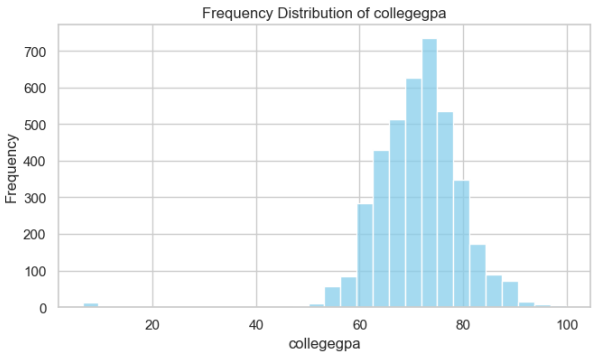

Probability Distribution of salary
1e−6
Probability Density
salary
1e6

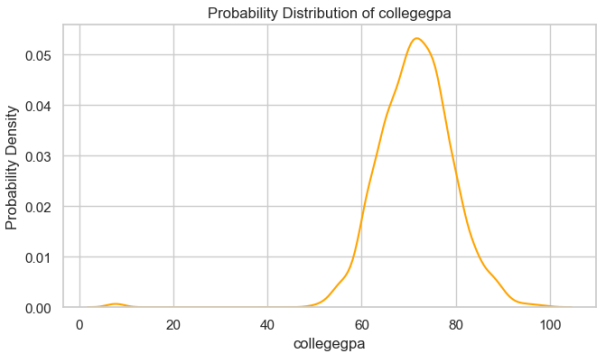

Frequency Distribution of 10percentage
Frequency
10percentage

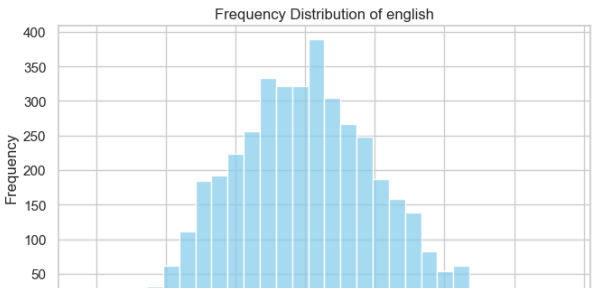

Probability Distribution of 10percentage
Probability Density
10percentage

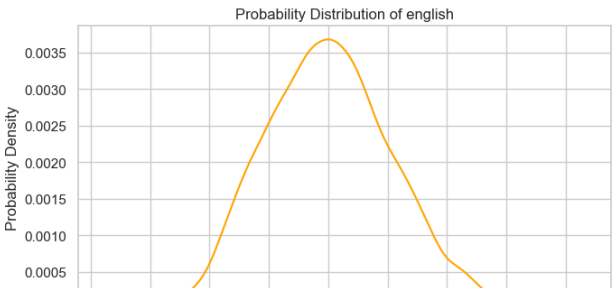

Frequency Distribution of 12percentage
Frequency
12percentage


Probability Distribution of 12percentage
Probability Density
12percentage


Frequency Distribution of collegegpa
Frequency
collegegpa

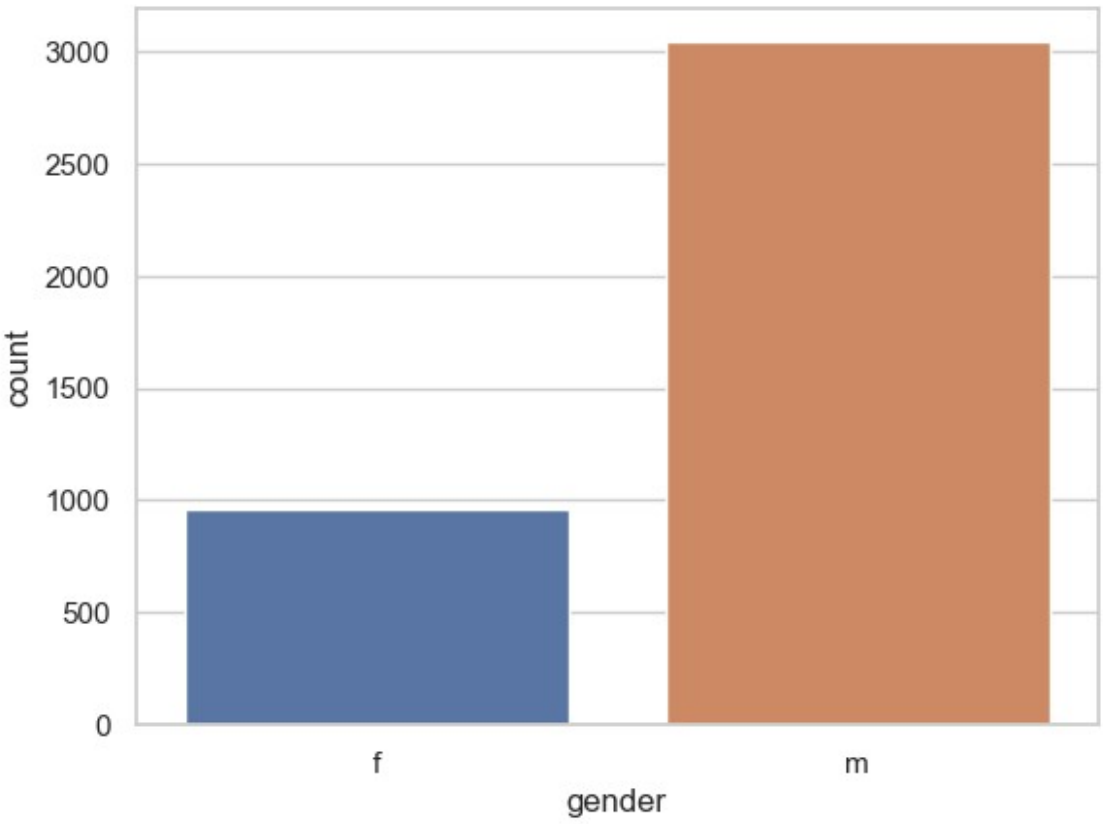

Probability Distribution of collegegpa
Probability Density
collegegpa


Frequency Distribution of english
Frequency


Probability Distribution of english
Probability Density

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

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

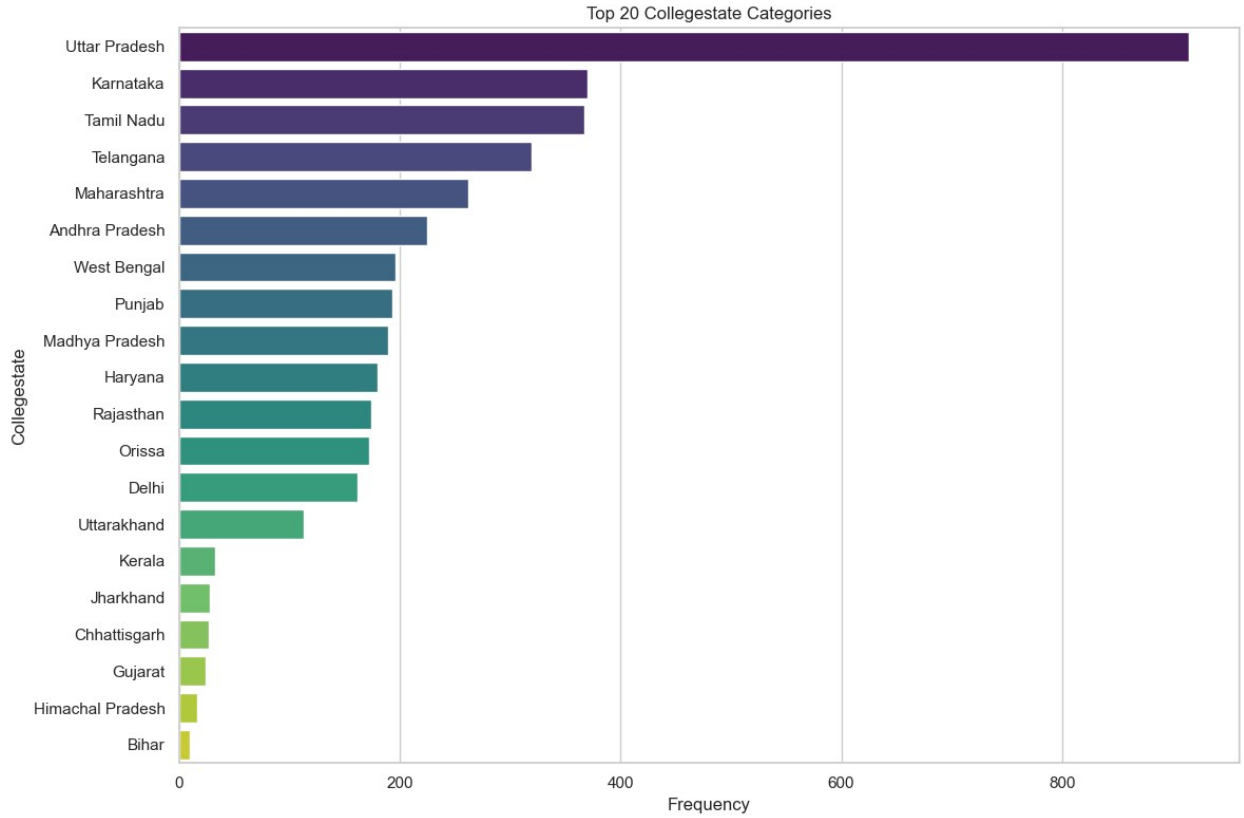



```
df.columns
```

```
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')
```

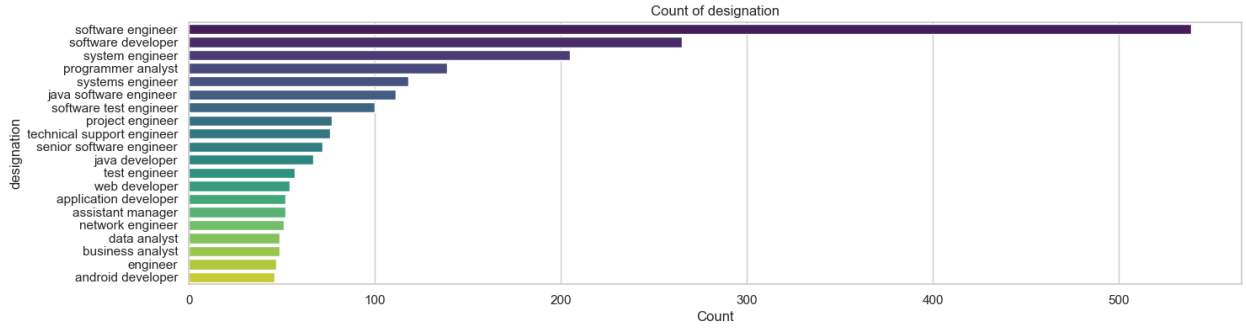
```
top_collegestates = df['collegestate'].value_counts().nlargest(20)
plt.figure(figsize=(12, 8))
```

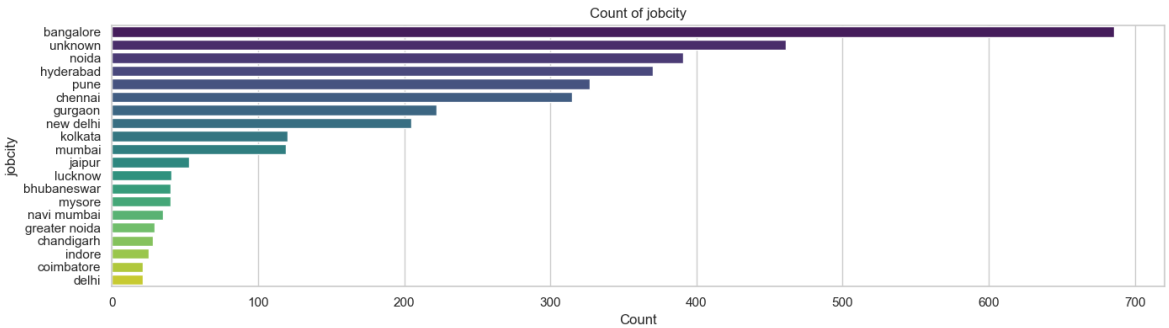
```
sns.countplot(
    y='collegestate',
    data=df[df['collegestate'].isin(top_collegestates.index)],
    palette='viridis',
    order=top_collegestates.index
)
plt.title('Top 20 Collegestate Categories')
plt.xlabel('Frequency')
plt.ylabel('Collegestate')
plt.tight_layout()
plt.show()
```

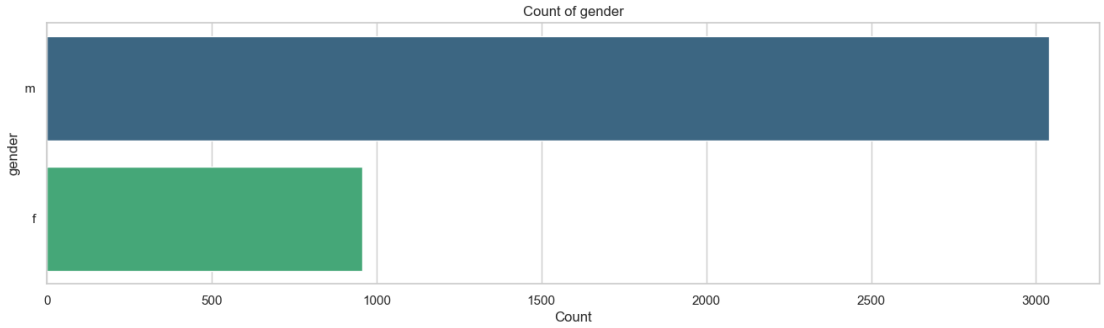


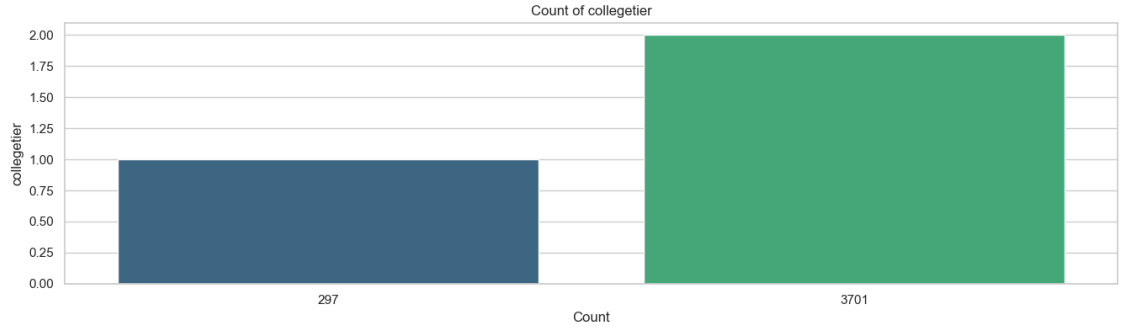

```
import matplotlib.pyplot as plt
import seaborn as sns

# 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
```

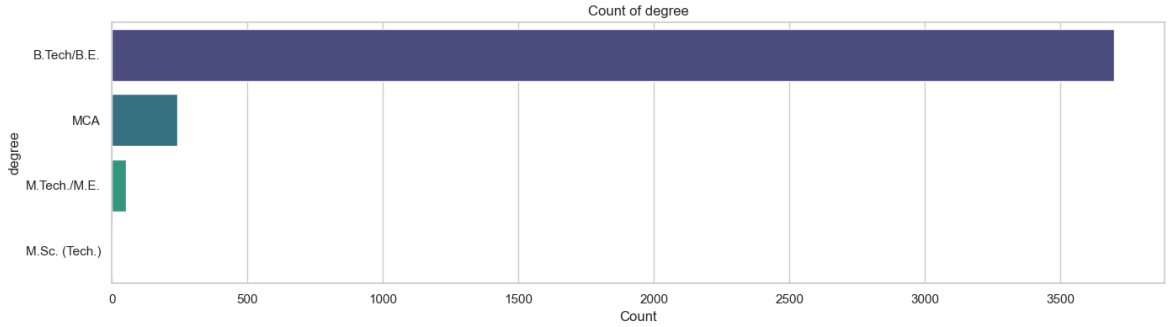
```
for i, column in enumerate(important_categorical_columns):
    plt.subplot(len(important_categorical_columns), 1, i + 1)  # 
Create a subplot for each column
    top_values = df[column].value_counts().nlargest(20)  # Get top 20 
values
    sns.barplot(x=top_values.values, y=top_values.index, 
palette='viridis')  # 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
project engineer
technical support engineer
senior software engineer
java developer
test engineer
web developer
application developer
assistant manager
network engineer
data analyst
business analyst
engineer
android developer
designation
0
100
200
300
400
500
Count

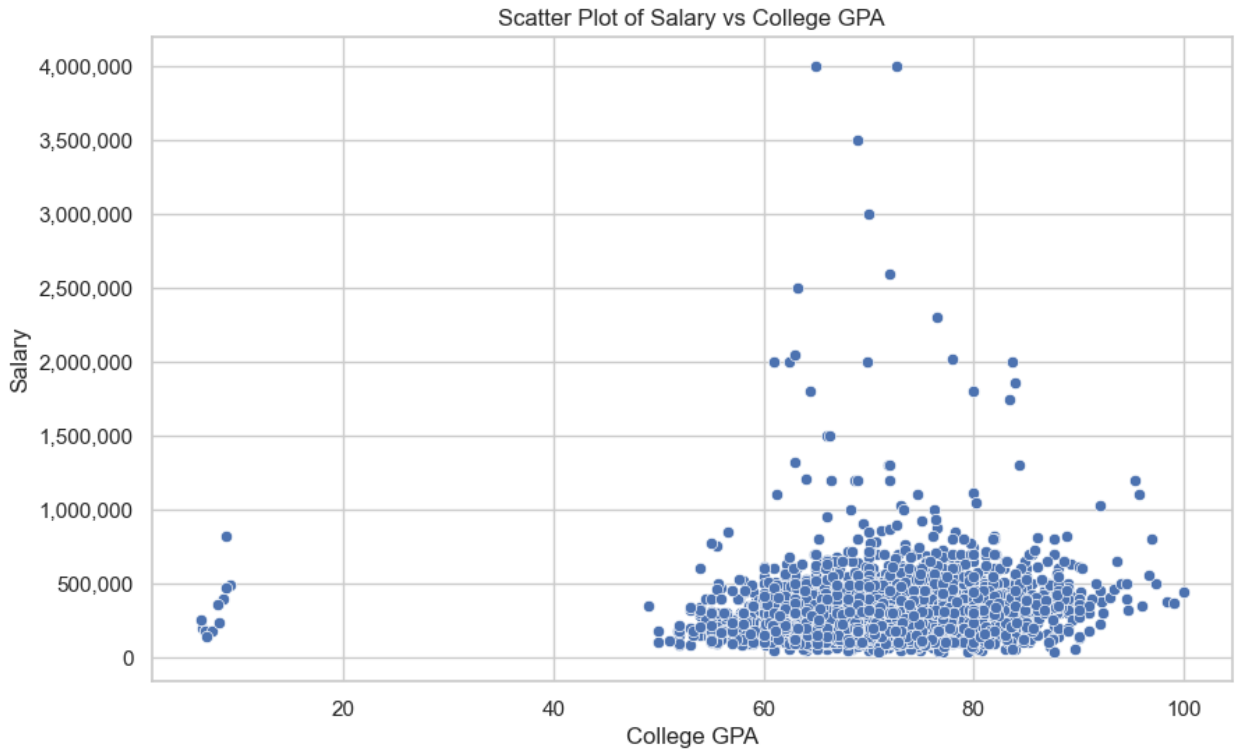

Count of jobcity
bangalore
unknown
noida
hyderabad
pune
chennai
gurgaon
new delhi
kolkata
mumbai
jaipur
lucknow
bhubaneswar
mysore
navi mumbai
greater noida
chandigarh
indore
coimbatore
delhi
jobcity
0
100
200
300
400
500
600
700
Count


Count of gender
m
f
gender
0
500
1000
1500
2000
2500
3000
Count


Count of collegetier
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
collegetier
297
3701
Count

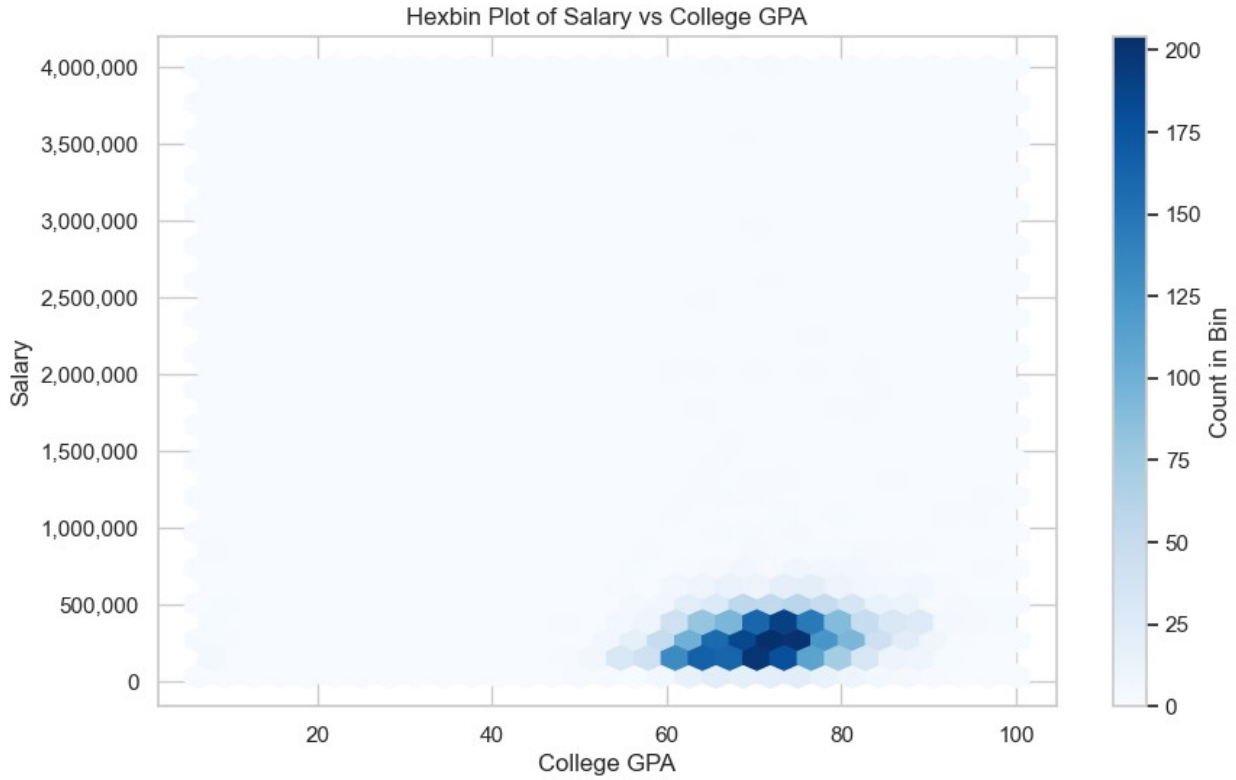

Count of degree
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)
degree
0
500
1000
1500
2000
2500
3000
3500
Count

```
from matplotlib.ticker import FuncFormatter
import matplotlib.pyplot as plt
import seaborn as sns

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

# Create the scatter plot
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))

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

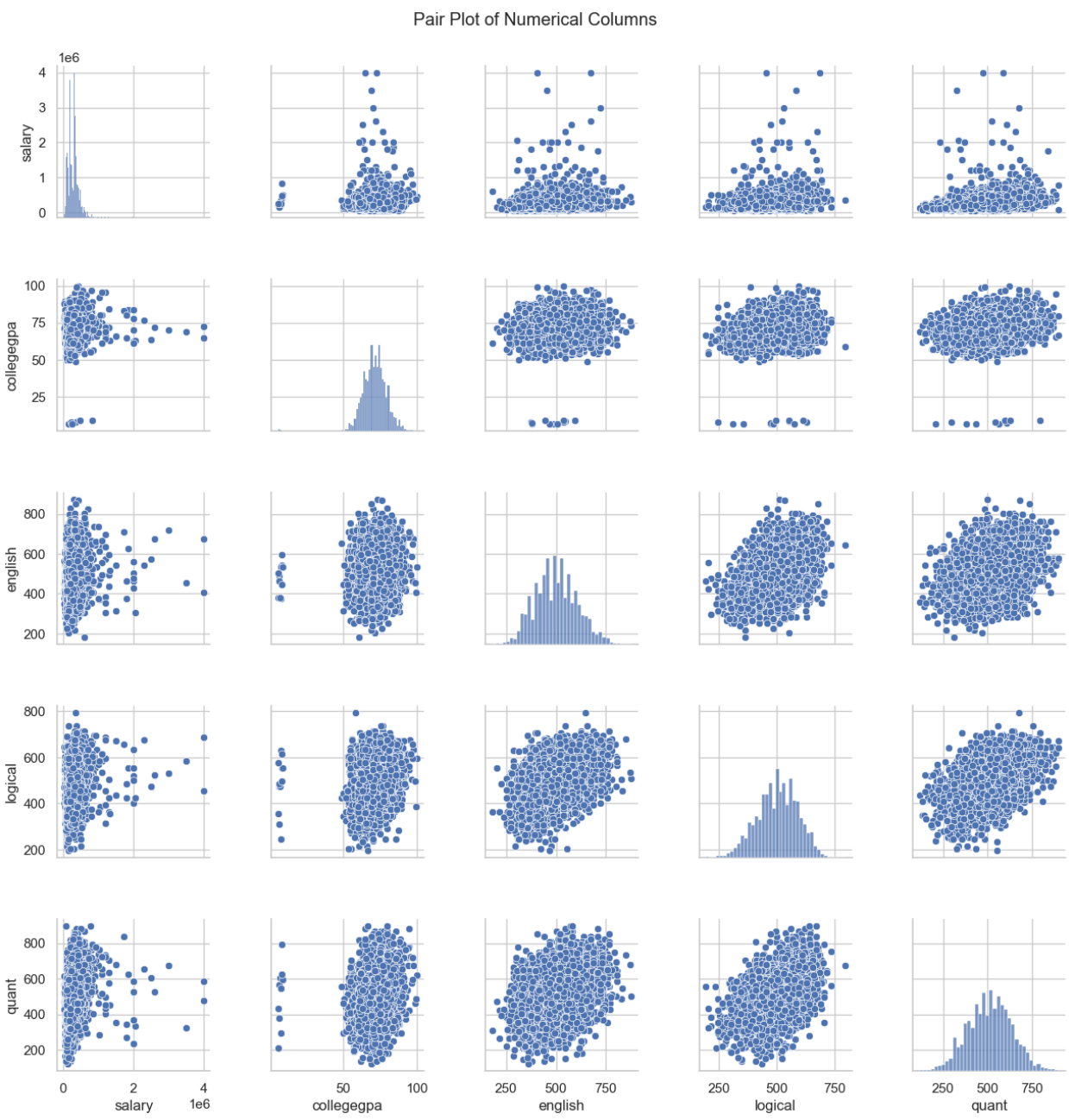



```
plt.figure(figsize=(10, 6))
plt.hexbin(df['collegegpa'], df['salary'], gridsize=30, cmap='Blues')
plt.colorbar(label='Count in Bin')
plt.title('Hexbin Plot of Salary vs College GPA')
```

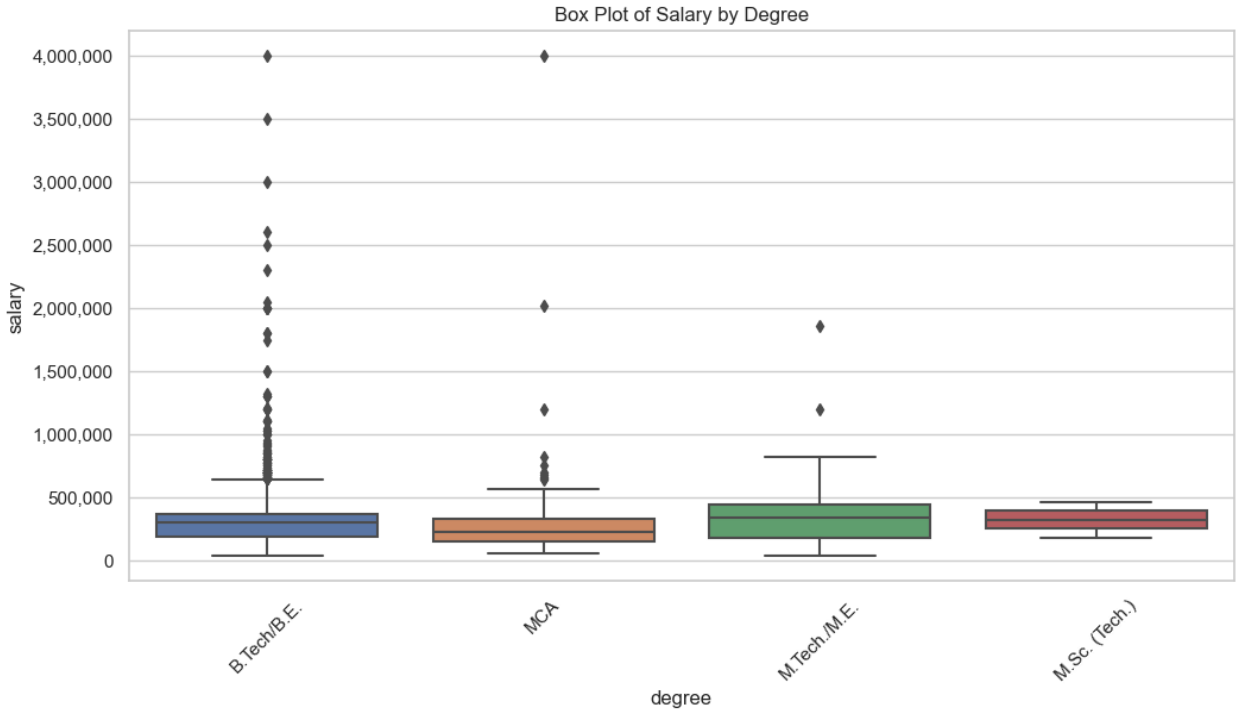
```
plt.xlabel('College GPA')
30
plt.ylabel('Salary')
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))
plt.show()
```



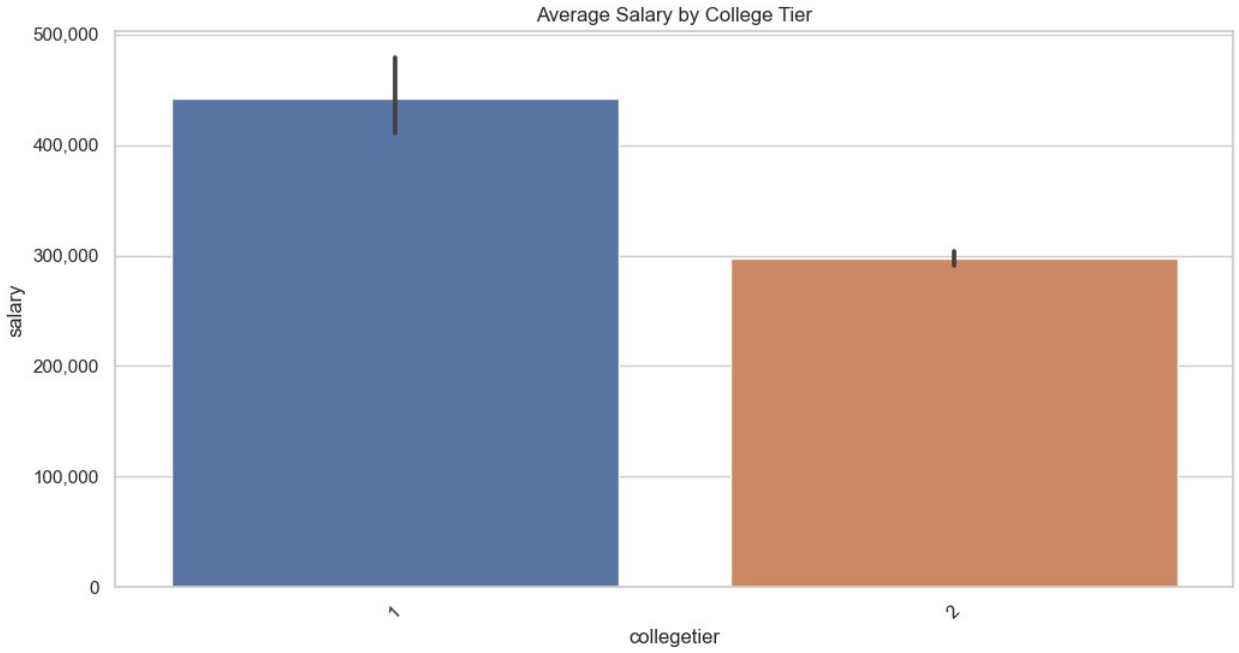

```
numerical_columns = ['salary', 'collegegpa', 'english', 'logical',
'quant']
sns.set(style="whitegrid")
pair_plot = sns.pairplot(df[numerical_columns])
plt.suptitle('Pair Plot of Numerical Columns', y=1.02)
plt.subplots_adjust(hspace=0.4, wspace=0.4)
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))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



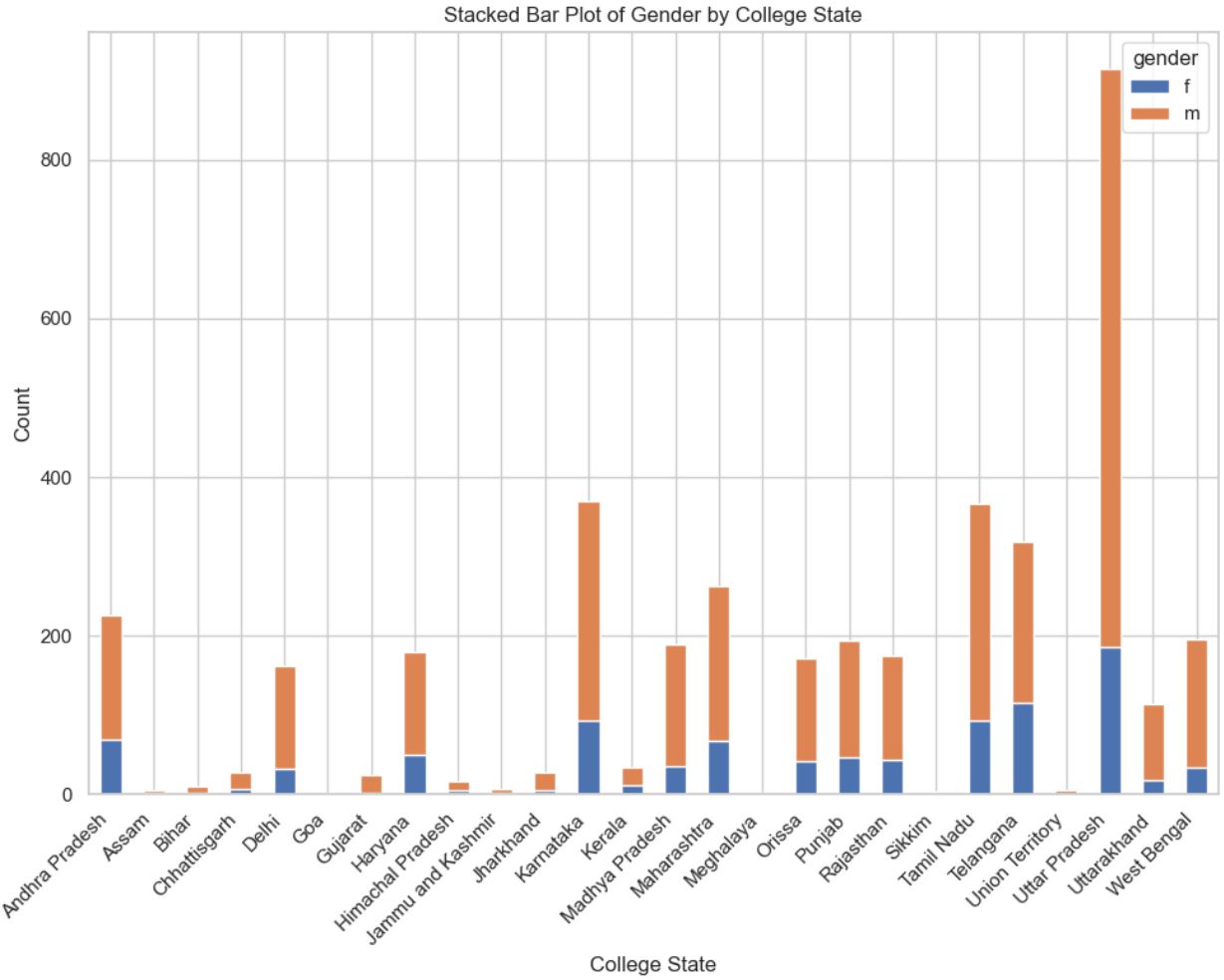

```
 plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='degree', y='salary')
plt.title('Box Plot of Salary by Degree')
plt.xticks(rotation=45)
plt.gca().yaxis.set_major_formatter(FuncFormatter(currency))
plt.show()
```




```
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()
```



```
import matplotlib.pyplot as plt

# Create a pivot table to count salaries by college state and gender
pivot_table = df.pivot_table(index='collegestate', columns='gender',
                             values='salary',
aggfunc='count').fillna(0)

# Plot a stacked bar plot
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()  # Show the plot
```



3 Step - 5 - 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 = 2.5 * 100000  # converting lakhs to actual number
upper_bound = 3 * 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()

# 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 * 100000 <= x <= 3 *
100000)

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: []
```

```
import pandas as pd
from scipy import stats

# 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)
```

```
  Cell In[56], line 10
    'Metric': ['Chi-Squared Statistic', 'P-value', 'Degrees of
Freedom',␣

^
SyntaxError: invalid character '␣' (U+2423)
```