INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

AMEO Data Analysis

Prepared by- **Soumen Mondal**

ABOUT ME

I am Soumen Mondal, a student of Computer Science & Design with a deep passion for data science. My academic journey is enriched by practical experiences, including my current internship at Innomatics Research Labs where I am applying my knowledge in real world scenarios.

My academic focus, my experiences extend beyond the classroom. My internships have provided me with a platform to apply my data science knowledge and computer science skills. My passion for uncovering insights from data and my eagerness to make impactful contributions are evident in my work. I am on a path to harness the power of data to drive decision-making and innovation.

LinkedIn : www.linkedin.com/in/soumen-2003-mondal
Github : msoumen097 (Soumen Mondal) (github.com)

INNOMATICS® RESEARCH LABS

I. Project Objective The primary goal of this analysis is to extract meaningful insights from the provided dataset, with a specific focus on understanding the relationship between different features and the target variable, which is Salary. The key objectives include:

•Thoroughly describing the dataset and its features.

•Identifying patterns or trends within the data.

•Exploring the relationships between independent variables and the target variable (Salary).

•Detecting any outliers or anomalies present in the dataset.

II. Data Overview The dataset under consideration is the Aspiring Mind Employment Outcome 2015 (AMEO) dataset, released by Aspiring Minds. It centers around employment outcomes for engineering graduates and encompasses various dependent variables such as Salary, Job Titles, and Job Locations. The dataset also includes standardized scores in cognitive skills, technical skills, and personality skills. With approximately 40 independent variables and 4000 data points, it comprises both continuous and categorical data. Additionally, demographic features and unique identifiers for each candidate are included.

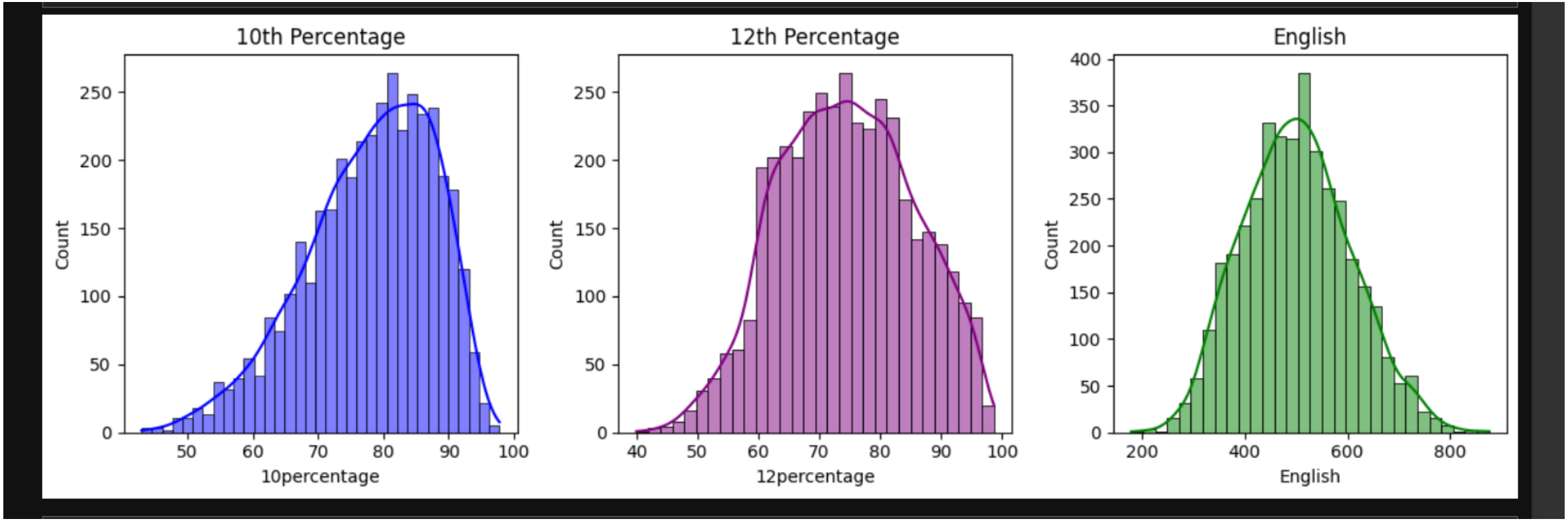
III. Data Cleaning and Preprocessing : A. Datatype Conversion To enhance the accuracy and consistency of the analysis, we converted the data types of 'Date of Joining' (DOJ) and 'Date of Leaving' (DOL) fields from their original format to datetime objects. Assuming the survey's 2015 date, respondents indicating 'present' for DOL were considered to have left the company by the latest survey date (2024-02-17).

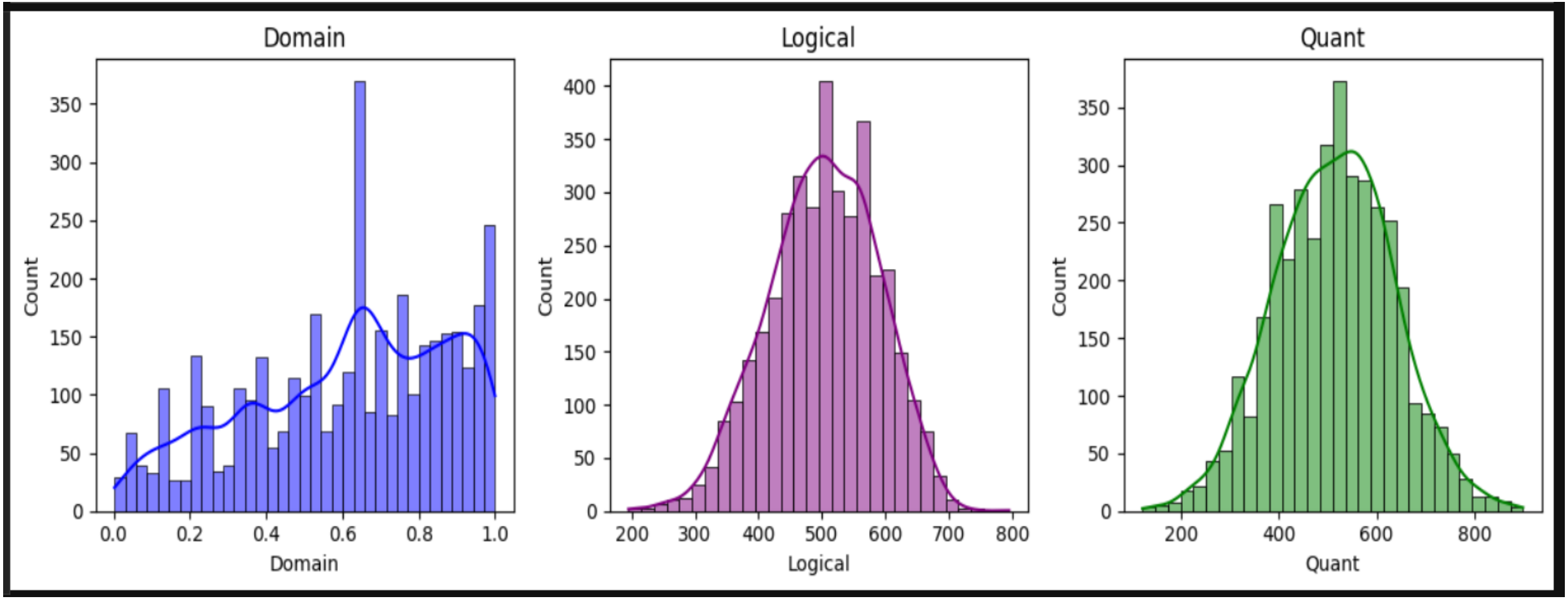
B. Validating 0 or -1 The initial data cleaning involved handling null values represented by 0 or -1. Columns such as '10board', '12board', 'GraduationYear', 'JobCity', and 'Domain' were processed. Subsequently, columns with over 80% -1 values, such as 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', and 'CivilEngg', were removed from further analysis. Optional subject columns, 'ElectronicsAndSemicon' and 'ComputerScience', had -1 values replaced with 0, indicating non-pursuit.

C. Collapsing Categories As a final step, the dataset was refined by retaining only the top 10 most frequent categories within specific columns. Categories beyond this selection were grouped as 'other,' streamlining the dataset for focused analysis.

❑ 10th percentage 12th percentage and CollegeGpa

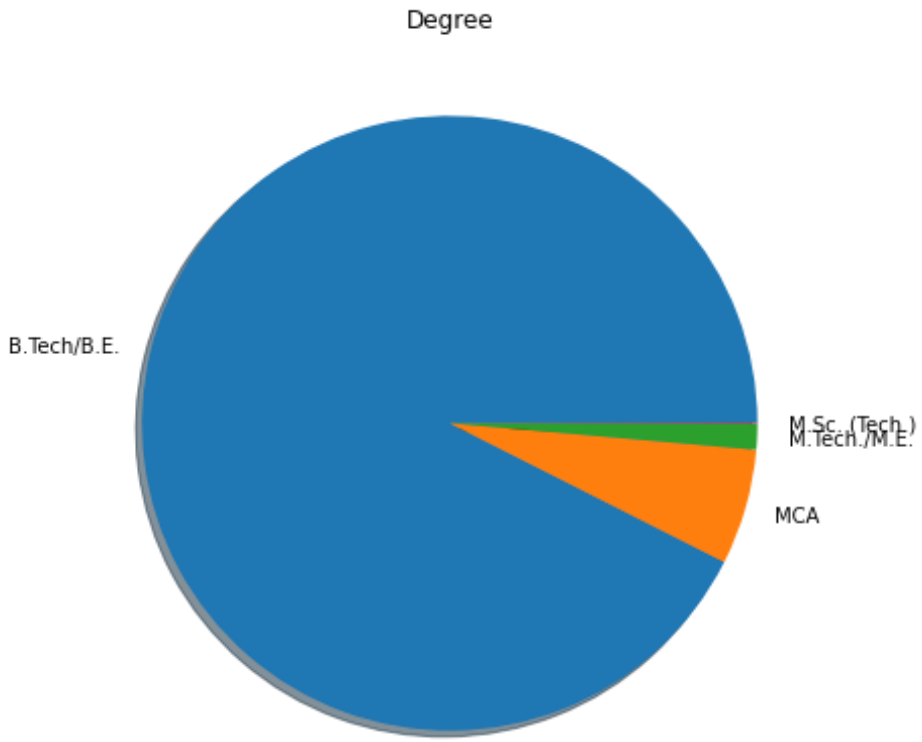
• In the univariate analysis, I concentrated on the 10th percentage, 12th percentage, and college CGPA. I plotted a graph to visualize the distribution of these variables. This analysis helped me understand the academic performance of the students and its impact on their employability. The graph provided insightful patterns and trends that are instrumental in understanding their influence on employability.



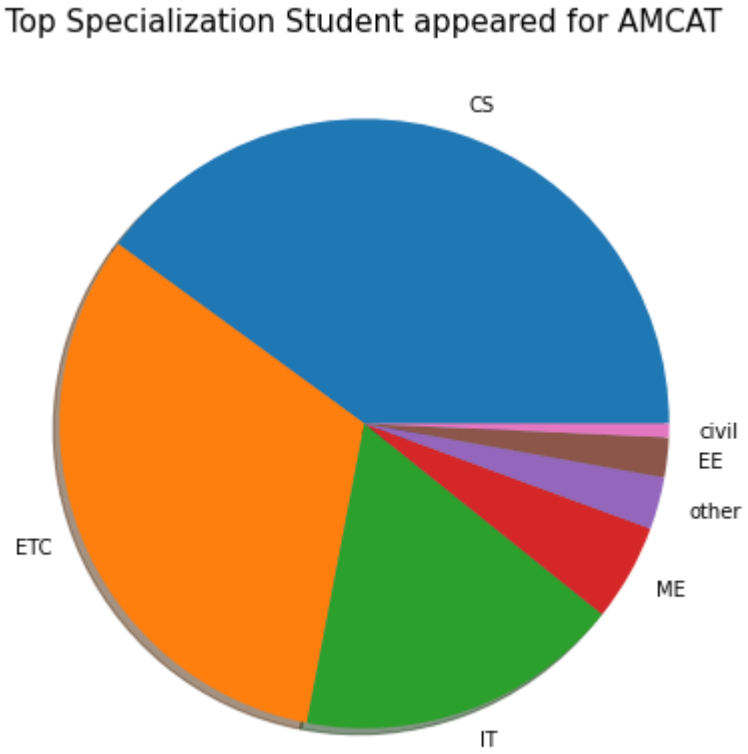


INNOMATICS®
RESEARCH LABS

UNIVARIANTE ANALYSIS

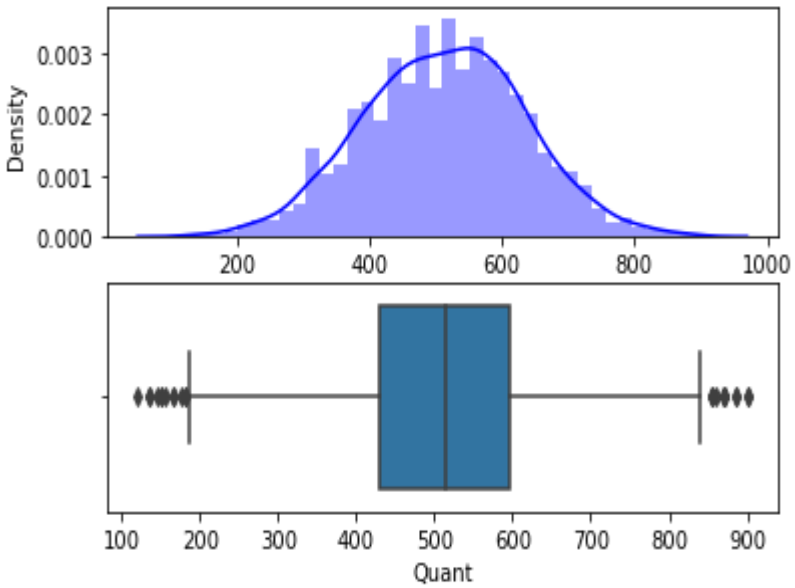


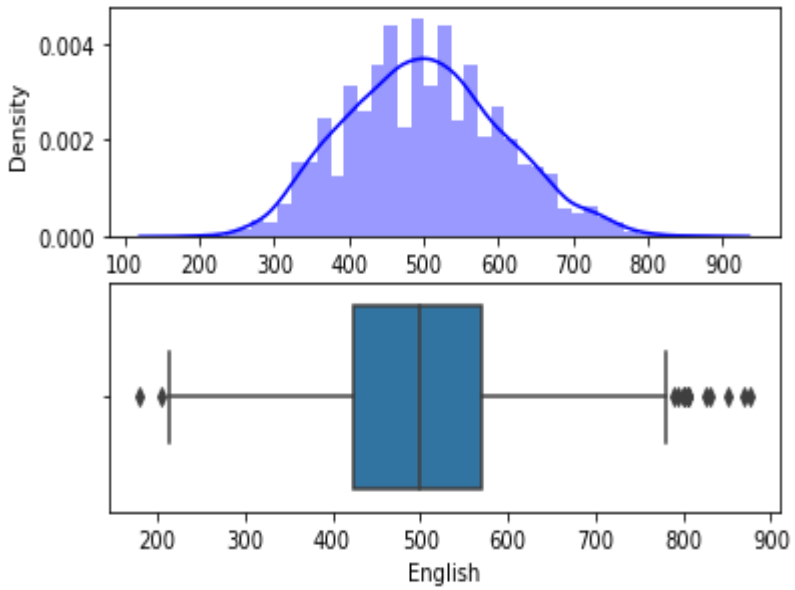
- B.Tech/B.E are more

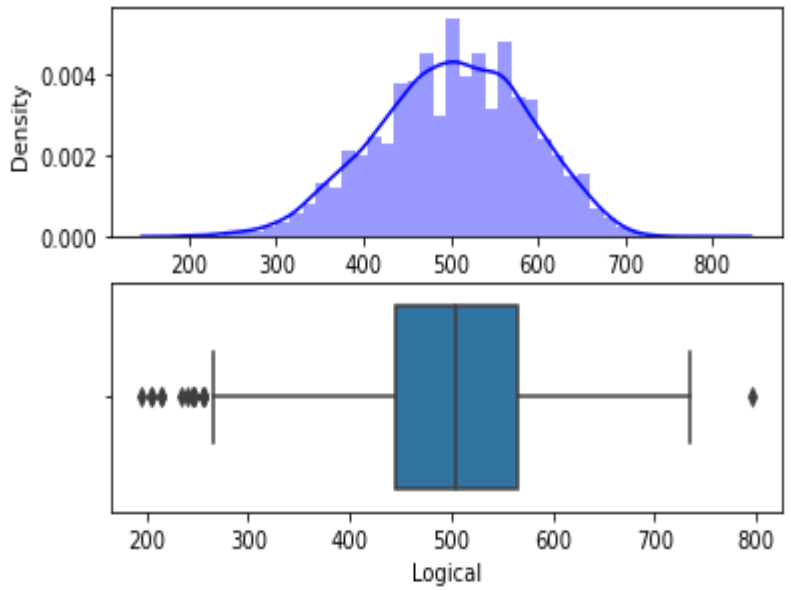


- CS, ETC and IT Branch student are giving more this exam

INNOMATICS® RESEARCH LABS







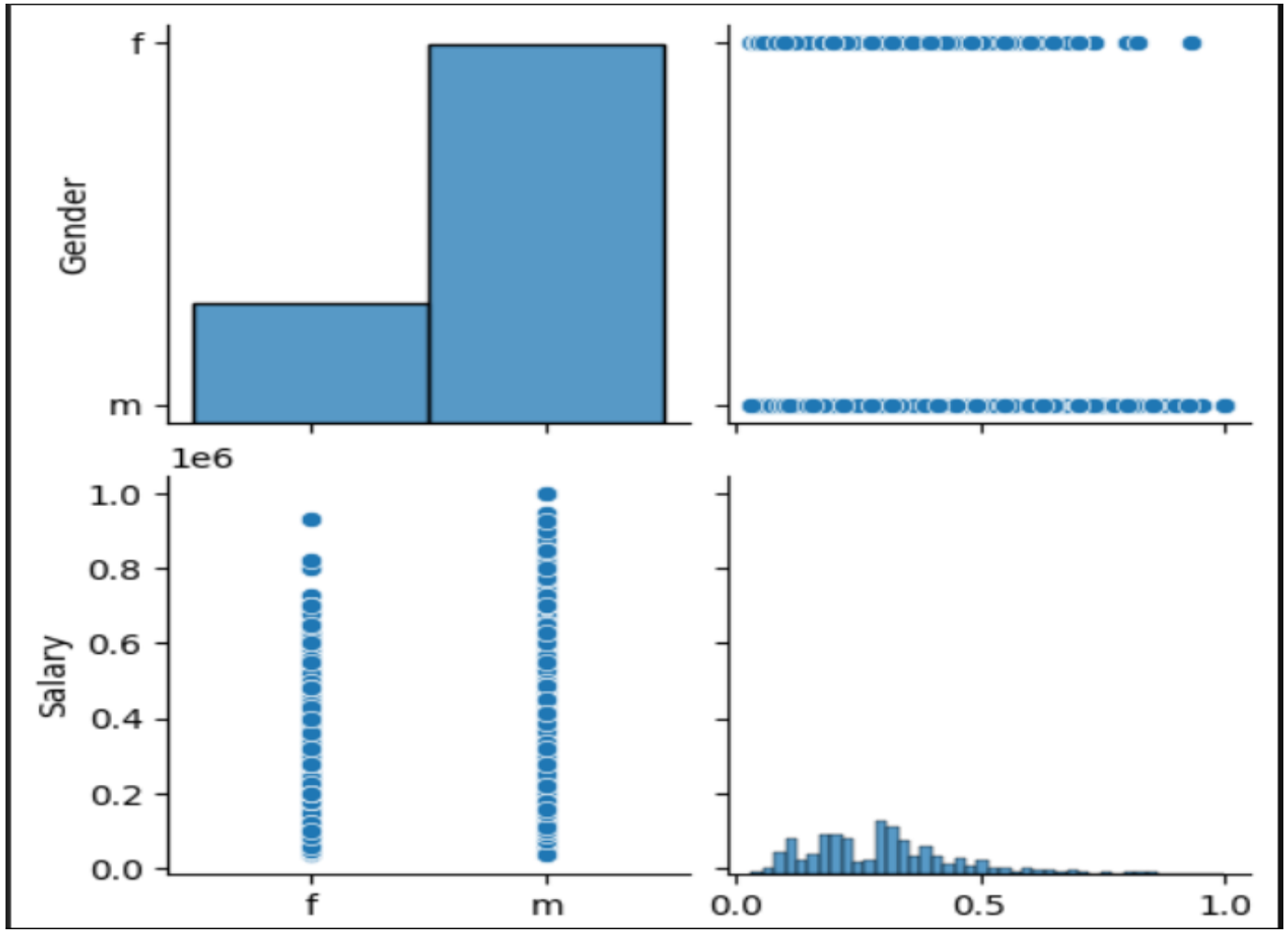
- The most of the quant range is between 430 to 595.
- The least value is 120.
- The highest value is 900

- The most of the english marks range is between 425 to 570.
- The least value is 180.
- The highest value is 875.

- The most of the logical marks range is between 445 to 565.
- The least value is 195.
- The highest value is 795

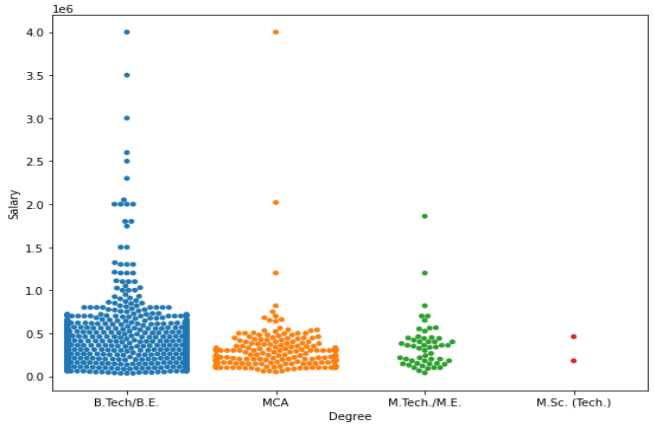
INNOMATİCS®
RESEARCH LABS

Males and females take the salary more or less the same

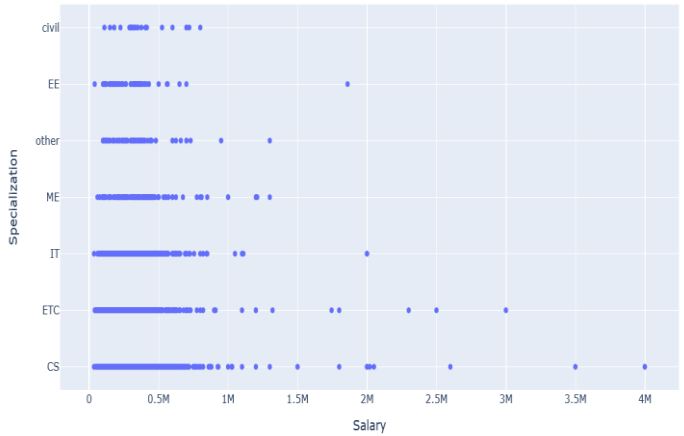


BIVARIANTE ANALYSIS

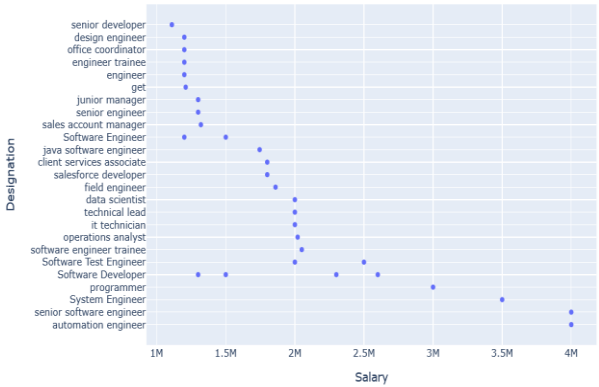
Analysis of Salary with various Variables



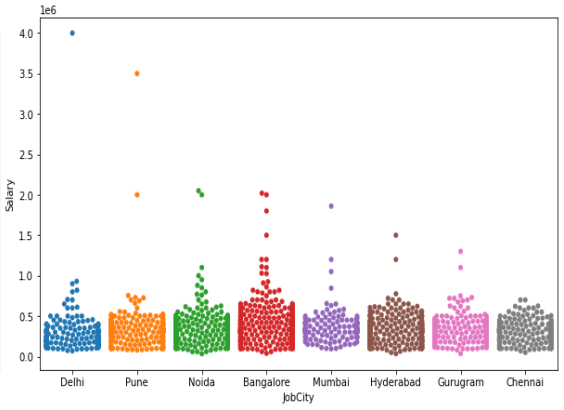
- B.Tech/BE and MCA has highest salary
- But Btech/BE gives wide range salary



- CS,IT and ETC branch candidate highest salary



- Roles that has the highest salary are:
  - Automation Engineer
  - Senior Software Engineer
  - System Engineer



- Delhi and Pune has the highest salary
- Noida, Bangalore and Hyderabad give wide range of salaries

INNOMATICS® RESEARCH LABS

CONCLUSION

- BTech /B.E and MCA has highest salary and highest joining in between 2010 to 2014
- CS,IT and ETC branch candidate highest salary
- Software engineer, Software test engineer, System engineer and Software developer roles has the highest salary
- Delhi and Pune has the highest salary. But Bangalore , Noida, Hyderabad, Pune have more no. of jobs available and give wide range of salaries
- 2014 has highest joining
- AMCAT exam with low and intermediate score also getting high salary packages

THANK YOU

INNOMATICS®
RESEARCH LABS

Analysis of AMEO Data

In [64]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('data.xlsx - Sheet1.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	ComputerScience	Me
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	...	-1	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	...	-1	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	...	-1	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	...	-1	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.0	...	-1	

5 rows × 39 columns

In [4]:
```
df = df.drop(columns={'Unnamed: 0'})
df
```

Out[4]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Computer
0	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.30	board ofsecondary education,ap	...	
1	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.40	cbse	...	
2	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.00	cbse	...	
3	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.60	cbse	...	
4	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.00	cbse	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3993	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/15/87 0:00	52.09	cbse	...	
3994	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/27/92 0:00	90.00	state board	...	
3995	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	bse,odisha	...	
3996	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	state board	...	
3997	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	cbse	...	

3998 rows × 38 columns

In [5]:
```
df.info()
```

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

After Performing 'df.info()' we can found that there has no null values in any columns

In [6]:
```
df1 = df.drop(columns = ['ID','CollegeID','CollegeCityID'])
df1.head()
```

Out[6]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12graduation	...	ComputerScier
0	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	board ofsecondary education,ap	2007	...	
1	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	cbse	2007	...	
2	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	cbse	2010	...	
3	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	cbse	2007	...	
4	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.0	cbse	2008	...	

5 rows × 35 columns

Datatypes Conversion

1. DOL - Date of Leaving.

- The survey was conducted back in 2015 and therefore making an

assumption that the respondents who responded as present for DOL actually left the company within 2015 only. So, we will replace present value in DOL with 2024-02-17. Then we convert the datetype of DOJ and DOL to datetime.

In [7]:
```
df1['DOL'] = df1['DOL'].str.replace('present','22/02/24')
```

In [8]:
```
df1['DOL'] = pd.to_datetime(df1['DOL'])
df1['DOJ'] = pd.to_datetime(df1['DOJ'])
df1.head()
```

Out[8]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12graduation	...	ComputerScienc
0	420000.0	2012-06-01	2024-02-22	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	board ofsecondary education,ap	2007	...	-1
1	500000.0	2013-09-01	2024-02-22	assistant manager	Indore	m	10/4/89 0:00	85.4	cbse	2007	...	-1
2	325000.0	2014-06-01	2024-02-22	systems engineer	Chennai	f	8/3/92 0:00	85.0	cbse	2010	...	-1
3	1100000.0	2011-07-01	2024-02-22	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	cbse	2007	...	-1
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	2/27/91 0:00	78.0	cbse	2008	...	-1

5 rows × 35 columns

In [9]:
```
df1.dtypes
```

Out[9]:
```
Salary                        float64
DOJ                    datetime64[ns]
DOL                    datetime64[ns]
Designation                    object
JobCity                        object
Gender                         object
DOB                            object
10percentage                  float64
10board                        object
12graduation                    int64
12percentage                  float64
12board                        object
CollegeTier                     int64
Degree                         object
Specialization                 object
collegeGPA                    float64
CollegeCityTier                 int64
CollegeState                   object
GraduationYear                  int64
English                         int64
Logical                         int64
Quant                           int64
Domain                        float64
ComputerProgramming             int64
ElectronicsAndSemicon           int64
ComputerScience                 int64
MechanicalEngg                  int64
ElectricalEngg                  int64
TelecomEngg                     int64
CivilEngg                       int64
conscientiousness             float64
agreeableness                 float64
extraversion                  float64
nueroticism                   float64
openess_to_experience         float64
dtype: object
```

Checking if the `DOL (Date of leaving)` is actually greater than `DOJ (Date of joining)`.

In [10]:
```
df1[df1['DOJ'] > df1['DOL']].shape
```

Out[10]:
```
(40, 35)
```

So, here we can found that there has `40 entries` might be typos and so we will drop those 40 rows.

In [11]:
```
df1 = df1.drop(df1[~(df1['DOL'] > df1['DOJ'])].index)
print(df1.shape)
```

```
(3943, 35)
```

In [13]:
```
df1.head()
```

Out[13]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12graduation	...	ComputerScienc
0	420000.0	2012-06-01	2024-02-22	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	board ofsecondary education,ap	2007	...	-1
1	500000.0	2013-09-01	2024-02-22	assistant manager	Indore	m	10/4/89 0:00	85.4	cbse	2007	...	-1
2	325000.0	2014-06-01	2024-02-22	systems engineer	Chennai	f	8/3/92 0:00	85.0	cbse	2010	...	-1
3	1100000.0	2011-07-01	2024-02-22	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	cbse	2007	...	-1
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	2/27/91 0:00	78.0	cbse	2008	...	-1

5 rows × 35 columns

In [25]:
```
df1.columns
```

Out[25]:
```
Index(['Salary', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DOB',
       '10percentage', '10board', '12graduation', '12percentage', '12board',
       'CollegeTier', 'Degree', 'Specialization', 'collegeGPA',
       'CollegeCityTier', 'CollegeState', 'GraduationYear', 'English',
       'Logical', 'Quant', 'Domain', 'ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
       'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

Check Percentages and CGPA are Valid or not.

In [34]:
```
print((df1['10percentage'] <=10).sum())
print((df1['12percentage'] <=10).sum())
print((df1['collegeGPA'] <=10).sum())
```

```
0
0
12
```

As we can see there are 12 rows that are not valid so we need to fix them

In [35]:
```
df1.loc[df1['collegeGPA']<=10,'collegeGPA'] = (df1.loc[df1['collegeGPA']<=10,'collegeGPA']/10)*100
```

In [37]:
```
print((df1['collegeGPA'] <=10).sum())
```

```
0
```

In [42]:
```
(df1==0).sum()[(df1==0).sum() > 0]
```

Out[42]:
```
CollegeCityTier    2761
GraduationYear        1
dtype: int64
```

In [53]:
```
(df1==-1).sum()[(df1==-1).sum()>0]
```

Out[53]:
```
Domain                   242
ComputerProgramming      861
ElectronicsAndSemicon   2815
ComputerScience         3060
MechanicalEngg          3708
ElectricalEngg          3789
TelecomEngg             3571
CivilEngg               3901
dtype: int64
```

In [54]:
```
df1 = df1.drop(columns = ['MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg'])
df1.head()
```

Out[54]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12graduation	...	Quant	Domain
0	420000.0	2012-06-01	2024-02-22	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	board ofsecondary education,ap	2007	...	525	0.635979
1	500000.0	2013-09-01	2024-02-22	assistant manager	Indore	m	10/4/89 0:00	85.4	cbse	2007	...	780	0.960603
2	325000.0	2014-06-01	2024-02-22	systems engineer	Chennai	f	8/3/92 0:00	85.0	cbse	2010	...	370	0.450877
3	1100000.0	2011-07-01	2024-02-22	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	cbse	2007	...	625	0.974396
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	2/27/91 0:00	78.0	cbse	2008	...	465	0.124502

5 rows × 31 columns

In [55]:
```
df1['10board'] = df1['10board'].replace({'0':np.nan})
df1['12board'] = df1['12board'].replace({'0':np.nan})
df1['GraduationYear'] = df1['GraduationYear'].replace({0:np.nan})
df1['JobCity'] = df1['JobCity'].replace({'-1':np.nan})
df1['Domain'] = df1['Domain'].replace({-1:np.nan})
df1['ElectronicsAndSemicon'] = df1['ElectronicsAndSemicon'].replace({-1:0})
df1['ComputerScience'] = df1['ComputerScience'].replace({-1:0})
df1['ComputerProgramming'] = df1['ComputerProgramming'].replace({-1:np.nan})
```

In [60]:
```
df1['10board'].fillna(df1['10board'].mode()[0], inplace = True)
df1['12board'].fillna(df1['12board'].mode()[0], inplace = True)
df1['GraduationYear'].fillna(df1['GraduationYear'].mode()[0], inplace = True)
df1['JobCity'].fillna(df1['JobCity'].mode()[0], inplace = True)

df1
```

Out[60]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12graduation	...	Quant
0	420000.0	2012-06-01	2024-02-22	senior quality engineer	Bangalore	f	2/19/90 0:00	84.30	board ofsecondary education,ap	2007	...	525
1	500000.0	2013-09-01	2024-02-22	assistant manager	Indore	m	10/4/89 0:00	85.40	cbse	2007	...	780
2	325000.0	2014-06-01	2024-02-22	systems engineer	Chennai	f	8/3/92 0:00	85.00	cbse	2010	...	370
3	1100000.0	2011-07-01	2024-02-22	senior software engineer	Gurgaon	m	12/5/89 0:00	85.60	cbse	2007	...	625
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	2/27/91 0:00	78.00	cbse	2008	...	465
...	...	...	...	...	...	...	...	...	...	...	...	...
3992	800000.0	2014-04-01	2015-04-01	manager	Rajkot	m	6/22/90 0:00	73.00	cbse	2008	...	525
3993	280000.0	2011-10-01	2012-10-01	software engineer	New Delhi	m	4/15/87 0:00	52.09	cbse	2006	...	475
3995	320000.0	2013-07-01	2024-02-22	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	bse,odisha	2008	...	465
3996	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	state board	2010	...	320
3997	400000.0	2013-02-01	2024-02-22	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	cbse	2008	...	464

3943 rows × 31 columns

In [57]:
```
df1['10board'] = df1['10board'].astype('category')
df1['12board'] = df1['12board'].astype('category')
df1['JobCity'] = df1['JobCity'].astype('category')
```

In [61]:
```
df1.info()
df1.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3943 entries, 0 to 3997
Data columns (total 31 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Salary                 3943 non-null   float64
 1   DOJ                    3943 non-null   datetime64[ns]
 2   DOL                    3943 non-null   datetime64[ns]
 3   Designation            3943 non-null   object
 4   JobCity                3943 non-null   category
 5   Gender                 3943 non-null   object
 6   DOB                    3943 non-null   object
 7   10percentage           3943 non-null   float64
 8   10board                3943 non-null   category
 9   12graduation           3943 non-null   int64
 10  12percentage           3943 non-null   float64
 11  12board                3943 non-null   category
 12  CollegeTier            3943 non-null   int64
 13  Degree                 3943 non-null   object
 14  Specialization         3943 non-null   object
 15  collegeGPA             3943 non-null   float64
 16  CollegeCityTier        3943 non-null   int64
 17  CollegeState           3943 non-null   object
 18  GraduationYear         3943 non-null   float64
 19  English                3943 non-null   int64
 20  Logical                3943 non-null   int64
 21  Quant                  3943 non-null   int64
 22  Domain                 3701 non-null   float64
 23  ComputerProgramming    3082 non-null   float64
 24  ElectronicsAndSemicon  3943 non-null   int64
 25  ComputerScience        3943 non-null   int64
 26  conscientiousness      3943 non-null   float64
 27  agreeableness          3943 non-null   float64
 28  extraversion           3943 non-null   float64
 29  nueroticism            3943 non-null   float64
 30  openess_to_experience  3943 non-null   float64
dtypes: category(3), datetime64[ns](2), float64(12), int64(8), object(6)
memory usage: 948.2+ KB
```

Out[61]:

```
Salary                   0
DOJ                      0
DOL                      0
Designation              0
JobCity                  0
Gender                   0
DOB                      0
10percentage             0
10board                  0
12graduation             0
12percentage             0
12board                  0
CollegeTier              0
Degree                   0
Specialization           0
collegeGPA               0
CollegeCityTier          0
CollegeState             0
GraduationYear           0
English                  0
Logical                  0
Quant                    0
Domain                 242
ComputerProgramming    861
ElectronicsAndSemicon    0
ComputerScience          0
conscientiousness        0
agreeableness            0
extraversion             0
nueroticism              0
openess_to_experience    0
dtype: int64
```

In [62]:

```
df1['Domain'].fillna(df1['Domain'].median(), inplace = True)
df1['ComputerProgramming'].fillna(df1['ComputerProgramming'].median(), inplace = True)
df1.head()
```

Out[62]:

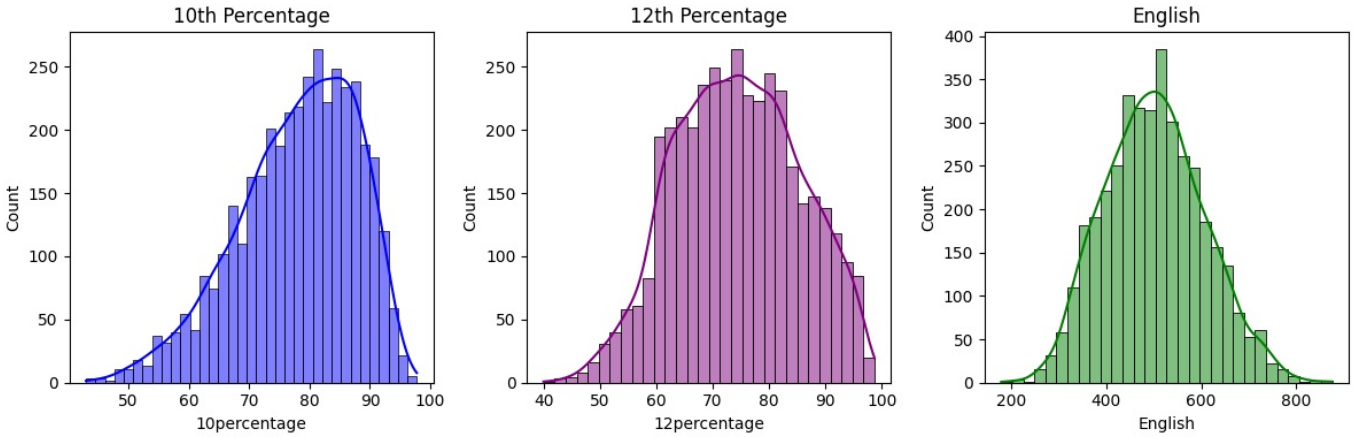
	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12graduation	...	Quant	Domain
0	420000.0	2012-06-01	2024-02-22	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	board ofsecondary education,ap	2007	...	525	0.635979
1	500000.0	2013-09-01	2024-02-22	assistant manager	Indore	m	10/4/89 0:00	85.4	cbse	2007	...	780	0.960603
2	325000.0	2014-06-01	2024-02-22	systems engineer	Chennai	f	8/3/92 0:00	85.0	cbse	2010	...	370	0.450877
3	1100000.0	2011-07-01	2024-02-22	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	cbse	2007	...	625	0.974396
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	2/27/91 0:00	78.0	cbse	2008	...	465	0.124502

5 rows × 31 columns

Univariate Analysis

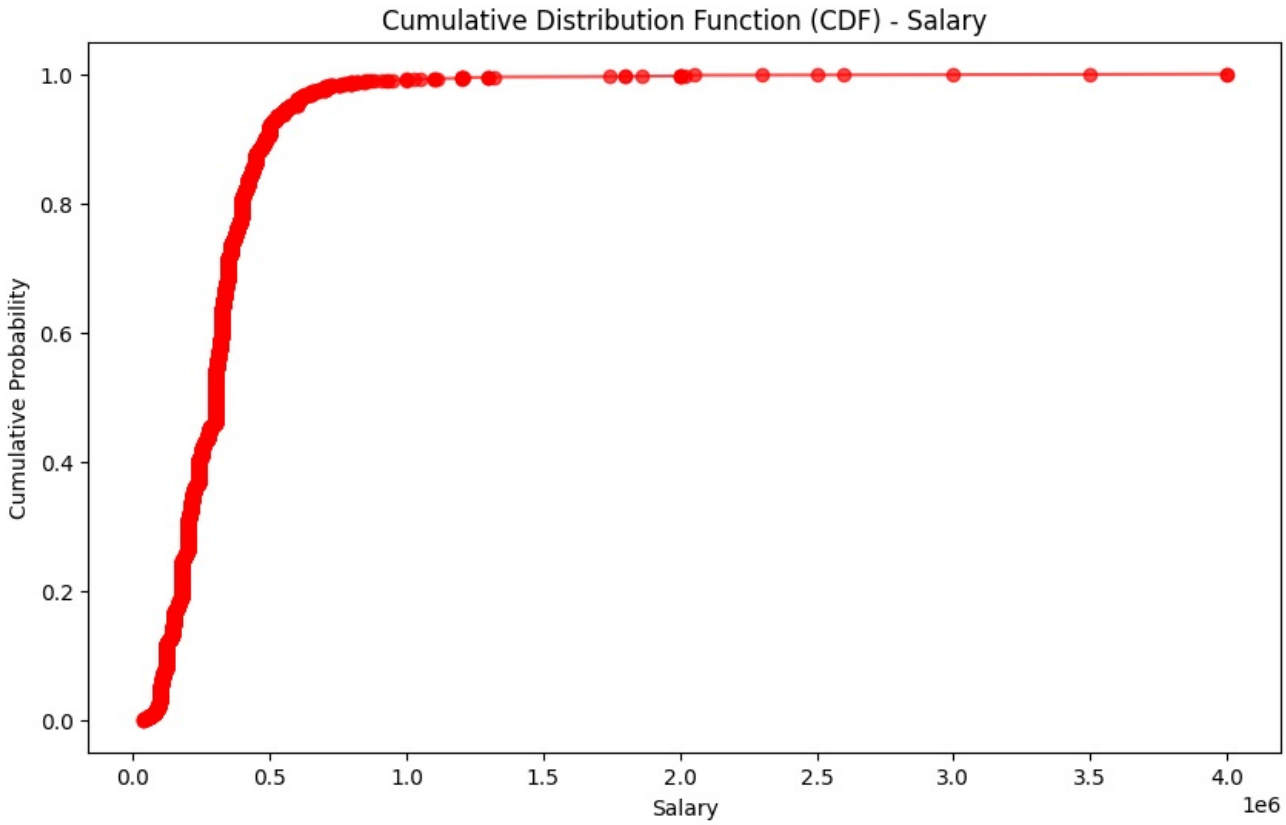
In [96]:

```
# Create subplots
plt.figure(figsize=(12, 4))

# Plot 1 - '10percentage'
plt.subplot(1, 3, 1)
sns.histplot(df1['10percentage'], bins=35, kde=True, color='blue')
plt.title('10th Percentage')

# Plot 2 - '12percentage'
plt.subplot(1, 3, 2)
sns.histplot(df1['12percentage'], bins=30, kde=True, color='purple')
plt.title('12th Percentage')

# Plot 2 - '12percentage'
plt.subplot(1, 3, 3)
sns.histplot(df1['English'], bins=30, kde=True, color='green')
plt.title('English')

# Adjust layout for better spacing
plt.tight_layout()

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



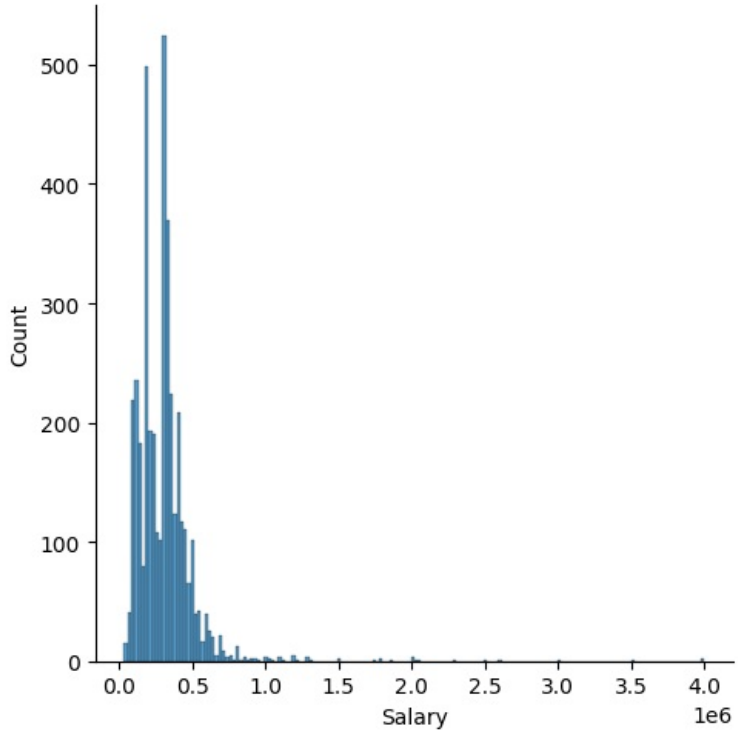
In [113…

```
sorted_salary = np.sort(df1['Salary'])

# Calculate the cumulative probabilities
cumulative_prob = np.arange(1, len(sorted_salary) + 1) / len(sorted_salary)

# Plot the CDF
plt.figure(figsize=(10, 6))
plt.plot(sorted_salary, cumulative_prob, marker='o',color='red',alpha = 0.7, label = 'Normal Distribution')
plt.title('Cumulative Distribution Function (CDF) - Salary')
plt.xlabel('Salary')
plt.ylabel('Cumulative Probability')
plt.show()
```



In [115…
```
sns.displot(df['Salary'])
```

Out[115…
```
<seaborn.axisgrid.FacetGrid at 0x235eee89890>
```



From above graph we can observe there is a outlier, Salary >10,00,000 is very rare, especially in the first job. So these are considered as outliers and removed.

In [117…
```
for i in range(1,8):
 seriesObj = df1.apply(lambda x: True if x['Salary'] <= 250000*i else False
, axis=1)
 # Count number of True in series
 numOfRows = len(seriesObj[seriesObj == True].index)
 print('Number of Rows in dataframe in which Salary %d : '%((250000*i)), numOfRows)
```

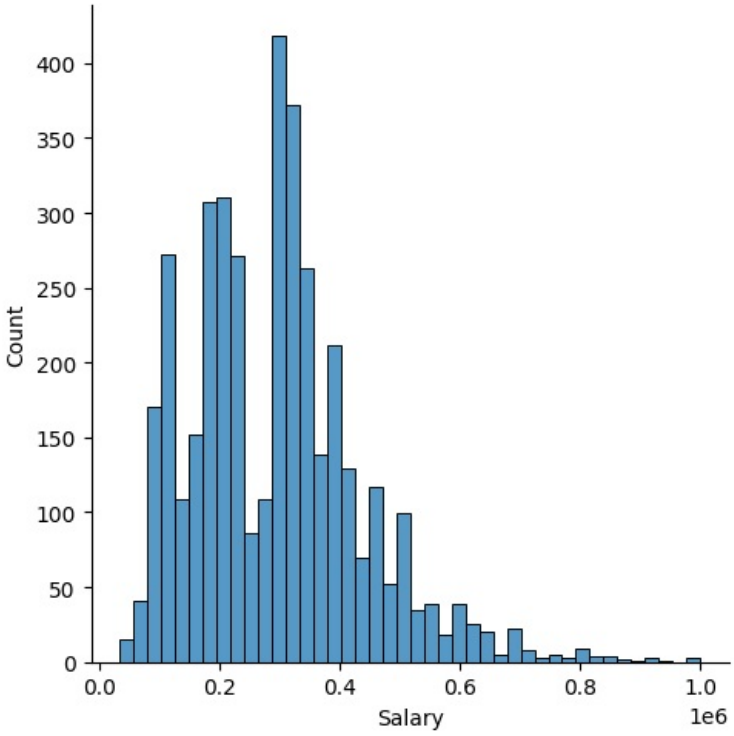
```
Number of Rows in dataframe in which Salary 250000 :  1628
Number of Rows in dataframe in which Salary 500000 :  3558
Number of Rows in dataframe in which Salary 750000 :  3799
Number of Rows in dataframe in which Salary 1000000 :  3831
Number of Rows in dataframe in which Salary 1250000 :  3843
Number of Rows in dataframe in which Salary 1500000 :  3847
Number of Rows in dataframe in which Salary 1750000 :  3848
```

In [119…
```
indexNames = df[ df['Salary'] > 1000000 ].index
# Delete these row indexes from dataFrame
df.drop(indexNames , inplace=True)
df.shape
```

Out[119… (3962, 38)

In [121…
```
sns.displot(df['Salary'])
```

Out[121… <seaborn.axisgrid.FacetGrid at 0x235c9b96590>



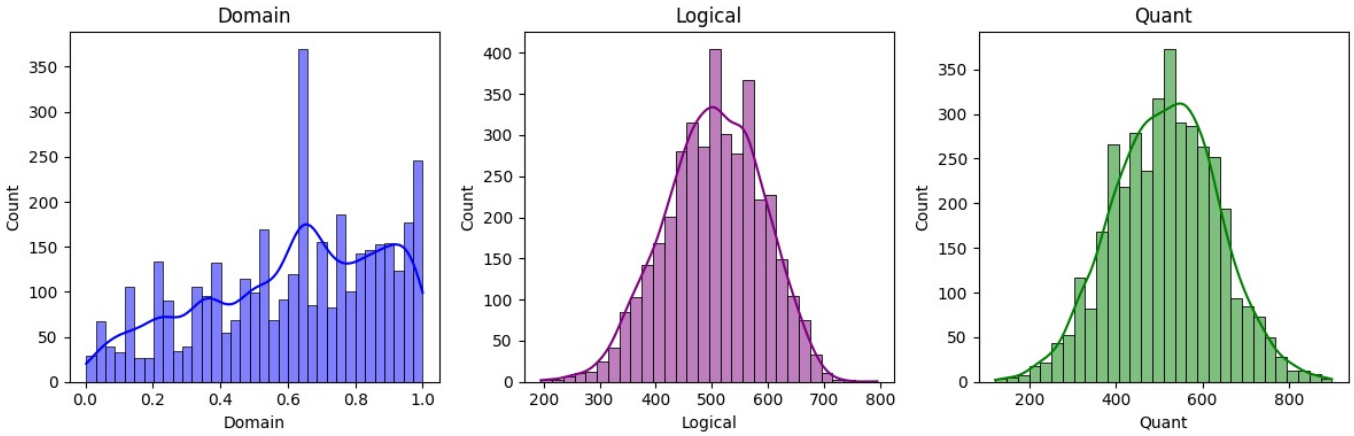
In [123…
```
df.describe().T
```

Out[123…

	count	mean	std	min	25%	50%	75%	
ID	3962.0	666103.566381	362228.136350	11244.0000	335343.500000	640177.000000	991153.250000	1.298275e
Salary	3962.0	294311.963655	141549.741483	35000.0000	180000.000000	300000.000000	365000.000000	1.000000e
10percentage	3962.0	77.926383	9.837770	43.0000	71.715000	79.030000	85.667500	9.776000e
12graduation	3962.0	2008.098688	1.644067	1995.0000	2007.000000	2008.000000	2009.000000	2.013000e
12percentage	3962.0	74.471108	11.003229	40.0000	66.000000	74.400000	82.600000	9.870000e
CollegeID	3962.0	5167.972993	4802.378568	2.0000	495.000000	3897.000000	8818.000000	1.840900e
CollegeTier	3962.0	1.927562	0.259245	1.0000	2.000000	2.000000	2.000000	2.000000e
collegeGPA	3962.0	71.470987	8.154160	6.4500	66.452500	71.710000	76.307500	9.993000e
CollegeCityID	3962.0	5167.972993	4802.378568	2.0000	495.000000	3897.000000	8818.000000	1.840900e
CollegeCityTier	3962.0	0.300606	0.458579	0.0000	0.000000	0.000000	1.000000	1.000000e
GraduationYear	3962.0	2012.110045	32.001278	0.0000	2012.000000	2013.000000	2014.000000	2.017000e
English	3962.0	501.506562	104.797696	180.0000	425.000000	500.000000	570.000000	8.750000e
Logical	3962.0	501.449268	86.572387	195.0000	445.000000	505.000000	565.000000	7.950000e
Quant	3962.0	513.262494	121.891206	120.0000	430.000000	515.000000	595.000000	9.000000e
Domain	3962.0	0.509997	0.468795	-1.0000	0.342315	0.622643	0.842248	9.999104
ComputerProgramming	3962.0	352.855881	205.289090	-1.0000	295.000000	415.000000	495.000000	8.400000e
ElectronicsAndSemicon	3962.0	95.354871	158.188989	-1.0000	-1.000000	-1.000000	233.000000	6.120000e
ComputerScience	3962.0	90.957597	175.442125	-1.0000	-1.000000	-1.000000	-1.000000	7.150000e
MechanicalEngg	3962.0	22.835184	97.947141	-1.0000	-1.000000	-1.000000	-1.000000	6.230000e
ElectricalEngg	3962.0	16.637557	87.966826	-1.0000	-1.000000	-1.000000	-1.000000	6.760000e
TelecomEngg	3962.0	31.950782	104.969246	-1.0000	-1.000000	-1.000000	-1.000000	5.480000e
CivilEngg	3962.0	2.717314	36.823026	-1.0000	-1.000000	-1.000000	-1.000000	5.160000e
conscientiousness	3962.0	-0.035956	1.027280	-4.1267	-0.649100	0.046400	0.702700	1.995300e
agreeableness	3962.0	0.144085	0.941614	-5.7816	-0.287100	0.212400	0.812800	1.904800e
extraversion	3962.0	0.000712	0.952281	-4.6009	-0.604800	0.091400	0.672000	2.535400e
nueroticism	3962.0	-0.165835	1.008124	-2.6430	-0.868200	-0.234400	0.526200	3.352500e
openess_to_experience	3962.0	-0.138433	1.008677	-7.3757	-0.669200	-0.094300	0.502400	1.822400e

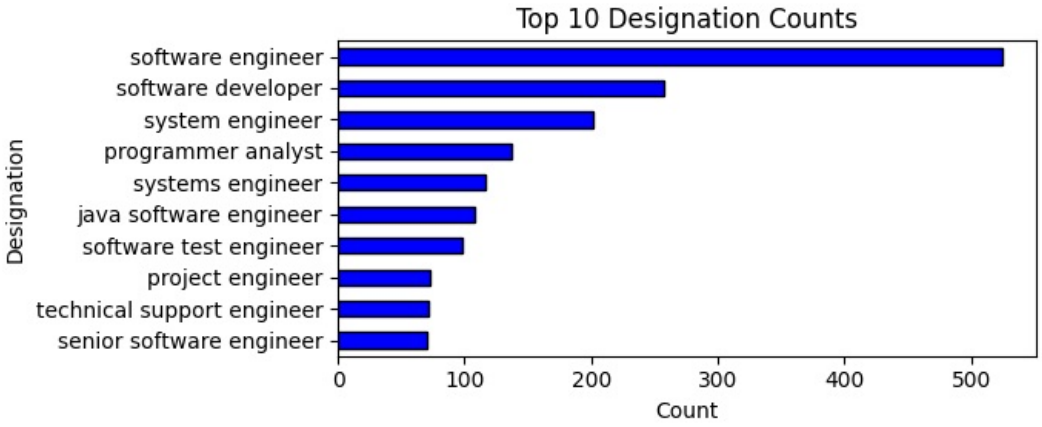
In [124…

```
# Create subplots
plt.figure(figsize=(12, 4))

# Plot 1 - '10percentage'
plt.subplot(1, 3, 1)
sns.histplot(df1['Domain'], bins=35, kde=True, color='blue')
plt.title('Domain')

# Plot 2 - '12percentage'
plt.subplot(1, 3, 2)
sns.histplot(df1['Logical'], bins=30, kde=True, color='purple')
plt.title('Logical')

# Plot 2 - '12percentage'
plt.subplot(1, 3, 3)
sns.histplot(df1['Quant'], bins=30, kde=True, color='green')
plt.title('Quant')

# Adjust layout for better spacing
plt.tight_layout()

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

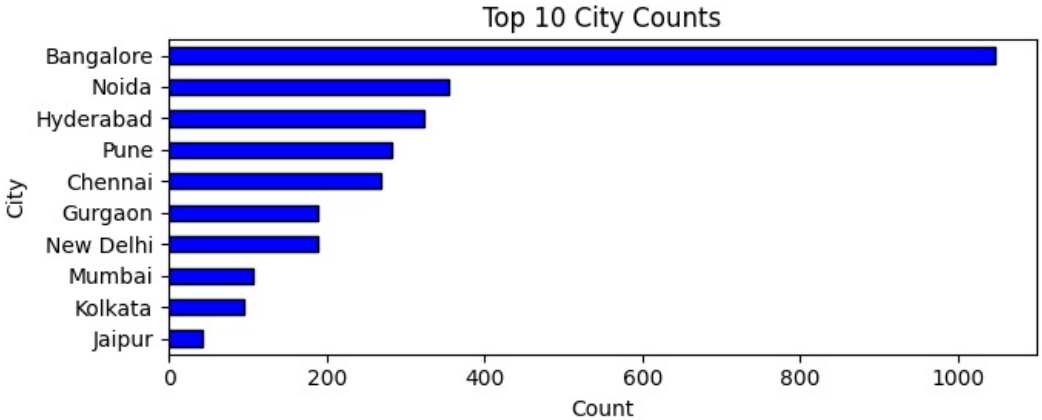


In [126…
```
categorical = ['Designation','JobCity', 'Gender','10board','12board','CollegeTier','Degree',
               'Specialization','CollegeCityTier','CollegeState']
for cat in categorical:
    df1[cat] = df1[cat].astype('category')
```

In [146…
```
# Get the top 10 designations by count
top_designations = df1['Designation'].value_counts().nlargest(10)

# Plot the bar chart for the top 10 designations
plt.figure(figsize=(7,3))
top_designations.sort_values().plot(kind='barh', color='blue', edgecolor='black')
plt.title('Top 10 Designation Counts')
plt.xlabel('Count')
plt.ylabel('Designation')
plt.tight_layout()
plt.show()
```

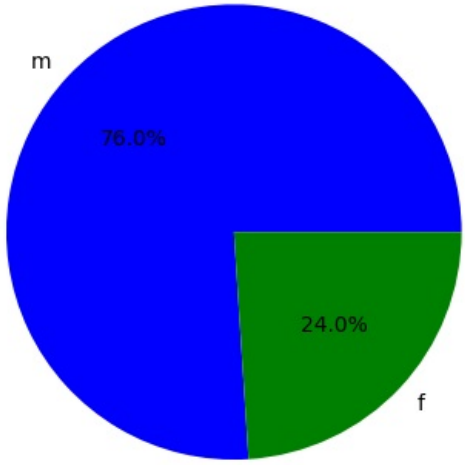


In [148…
```
# Get the top 10 designations by count
top_city = df1['JobCity'].value_counts().nlargest(10)

# Plot the bar chart for the top 10 designations
plt.figure(figsize=(7,3))
top_city.sort_values().plot(kind='barh', color='blue', edgecolor='black')
plt.title('Top 10 City Counts')
plt.xlabel('Count')
plt.ylabel('City')
plt.tight_layout()
plt.show()
```

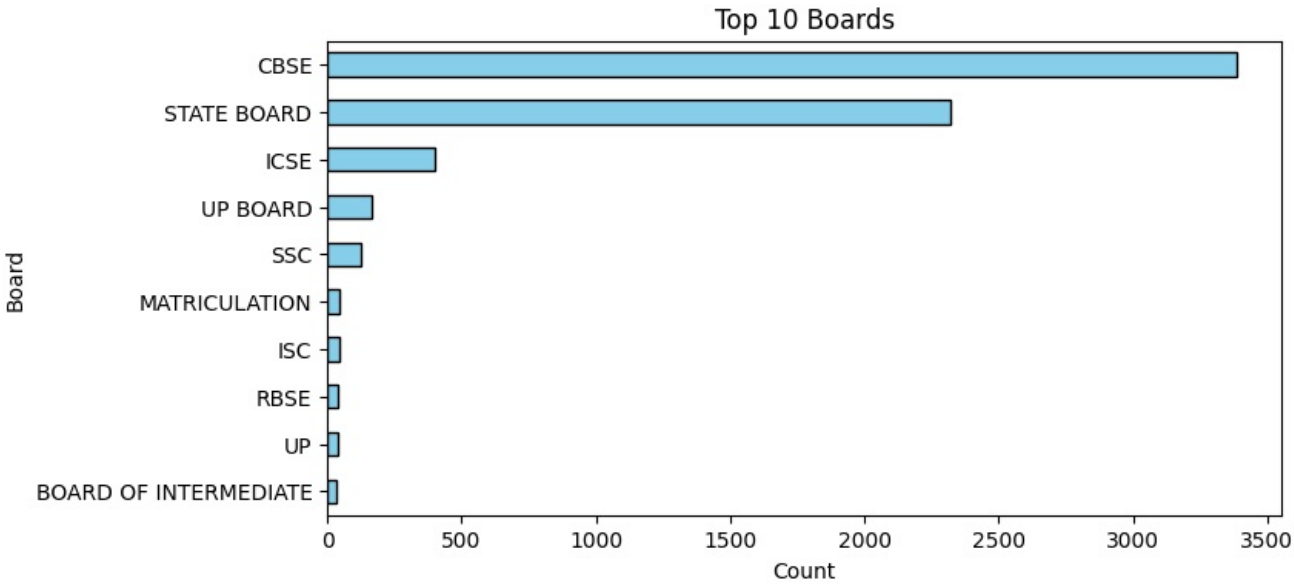


In [160…
```
gender_counts = df['Gender'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, autopct='%0.01f%%', colors=['blue', 'green'])
plt.plot()
```

Out[160… []

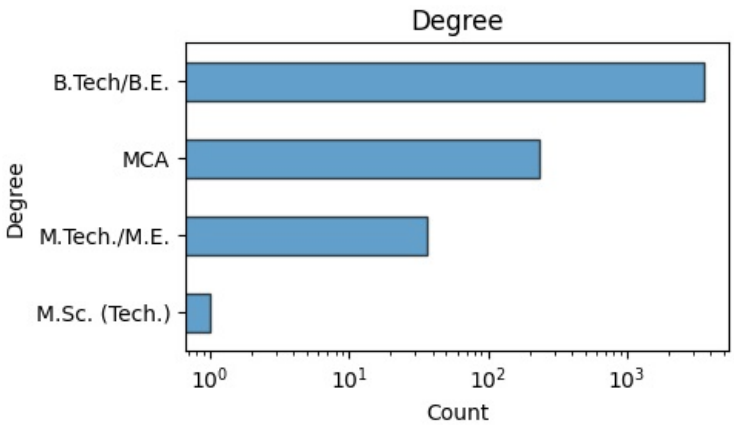


In [164…
```
import matplotlib.pyplot as plt

# Assuming df1 is your DataFrame with '10board' and '12board' columns

# Combine '10board' and '12board' data into a single DataFrame
combined_board_data = pd.concat([df1['10board'], df1['12board']])

# Convert board names to uppercase for consistency
combined_board_data = combined_board_data.str.upper()

# Count the occurrences of each board
board_counts = combined_board_data.value_counts()

# Plot the top N boards
top_n_boards = 10
plt.figure(figsize=(8, 4))
board_counts.nlargest(top_n_boards).sort_values().plot(kind='barh', color='skyblue', edgecolor='black')
plt.title(f'Top {top_n_boards} Boards')
plt.xlabel('Count')
plt.ylabel('Board')
plt.show()
```



In [167…
```
df1['Degree'].value_counts().sort_values(ascending=True).plot(
    kind='barh',
    title='Degree',
    figsize=(5, 3),
    ec='k',
    alpha=0.7
)
plt.ylabel('Degree')
```

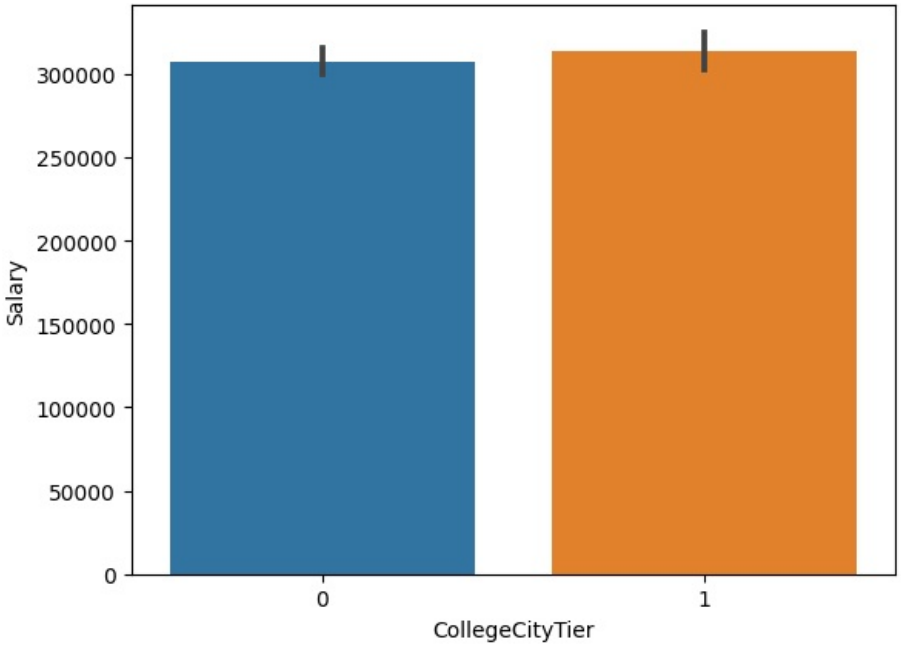
```
plt.xlabel('Count')
plt.xscale('log')
plt.tight_layout()
plt.show()
```



Bivariate analysis

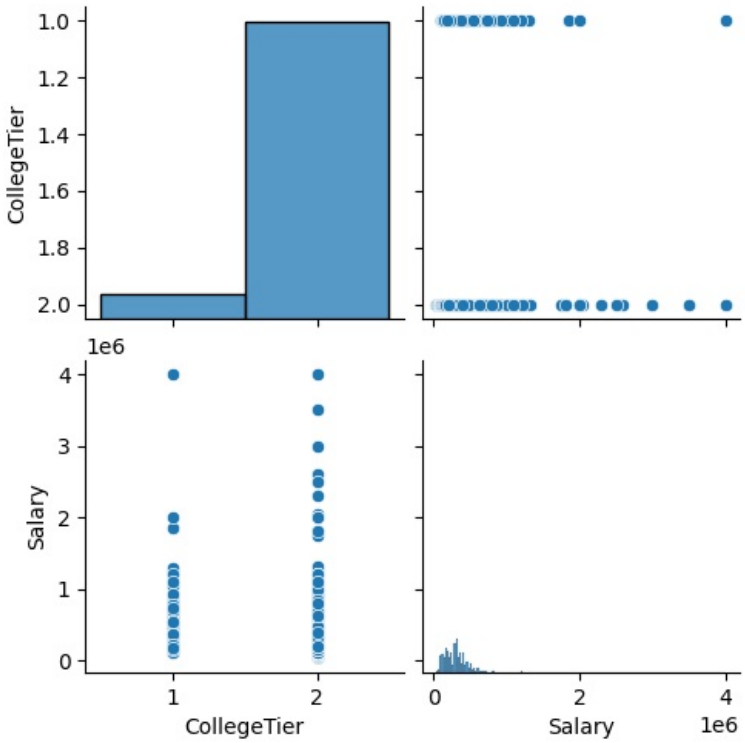
In [168…
```
sns.barplot(x='CollegeCityTier',y='Salary',data=df1)
```

Out[168…
```
<AxesSubplot: xlabel='CollegeCityTier', ylabel='Salary'>
```



In [169…
```
sns.pairplot(df1,vars=['CollegeTier', 'Salary'])
```

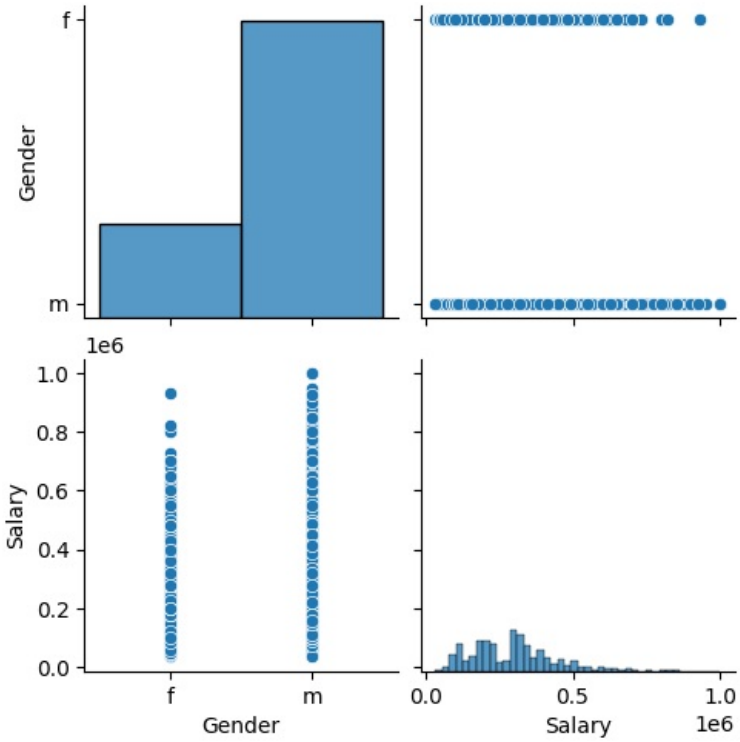
Out[169…
```
<seaborn.axisgrid.PairGrid at 0x235f5ca70d0>
```



From the graph we observe that collegecitytier 1 has bagged with highest salary , and also to be noted that collegecity tier 0 also provide the same salary expectation

In [175…
```
sns.pairplot(df,vars=['Gender', 'Salary']);

print('Males and females take the salary more or less the same')
```

```
Males and females take the salary more or less the same
```



In [180…

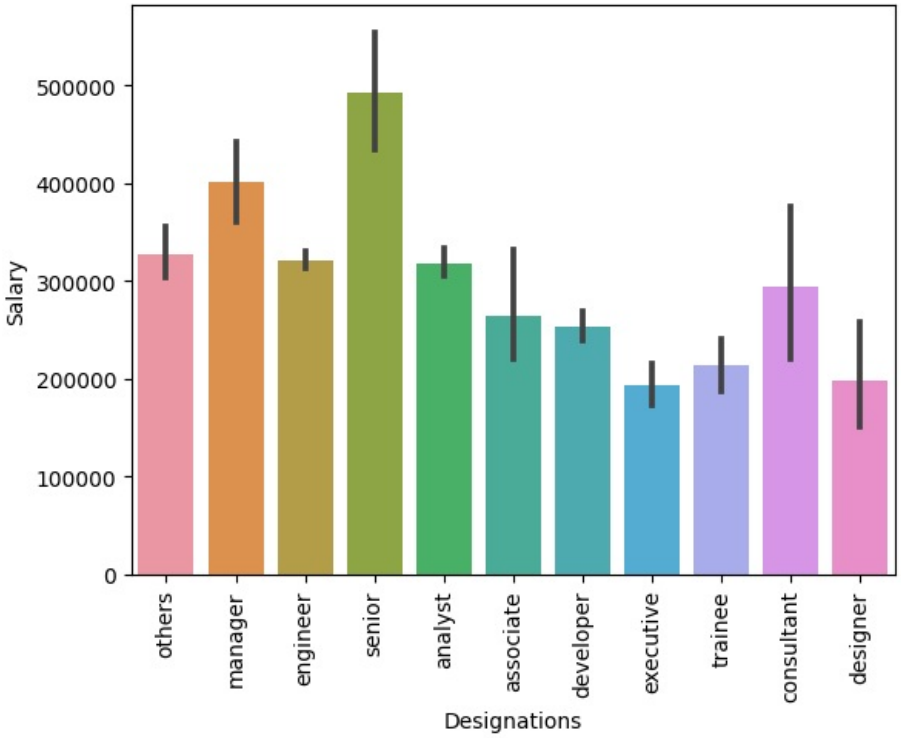
```
l=[]
for i in df1['Designation']:
 if ('senior' in i and 'engineer' not in i):
     l.append('senior')
 elif('trainee'in i and 'engineer' not in i):
     l.append('trainee')
 elif('engineer' in i and 'senior' not in i):
     l.append('engineer')
 elif('associate' in i and 'senior' not in i):
     l.append('associate')
 elif('developer' in i and 'senior' not in i):
     l.append('developer')
 elif('manager' in i and 'senior' not in i):
     l.append('manager')
 elif('analyst' in i):
     l.append('analyst')
 elif('consultant' in i):
     l.append('consultant')
 elif('executive' in i):
     l.append('executive')
 elif('designer' in i):
     l.append('designer')
 else:
     l.append('others')
```

In [183…

```
df1['Designations']=l
df1['Designations'].value_counts()
```

Out[183…
```
engineer      1928
developer      647
others         521
analyst        390
manager        116
associate       64
executive       58
trainee         56
senior          42
designer        23
consultant      19
Name: Designations, dtype: int64
```

In [184…
```
sns.barplot(x='Designations',y='Salary',data=df1)
plt.xticks(rotation=90);
```

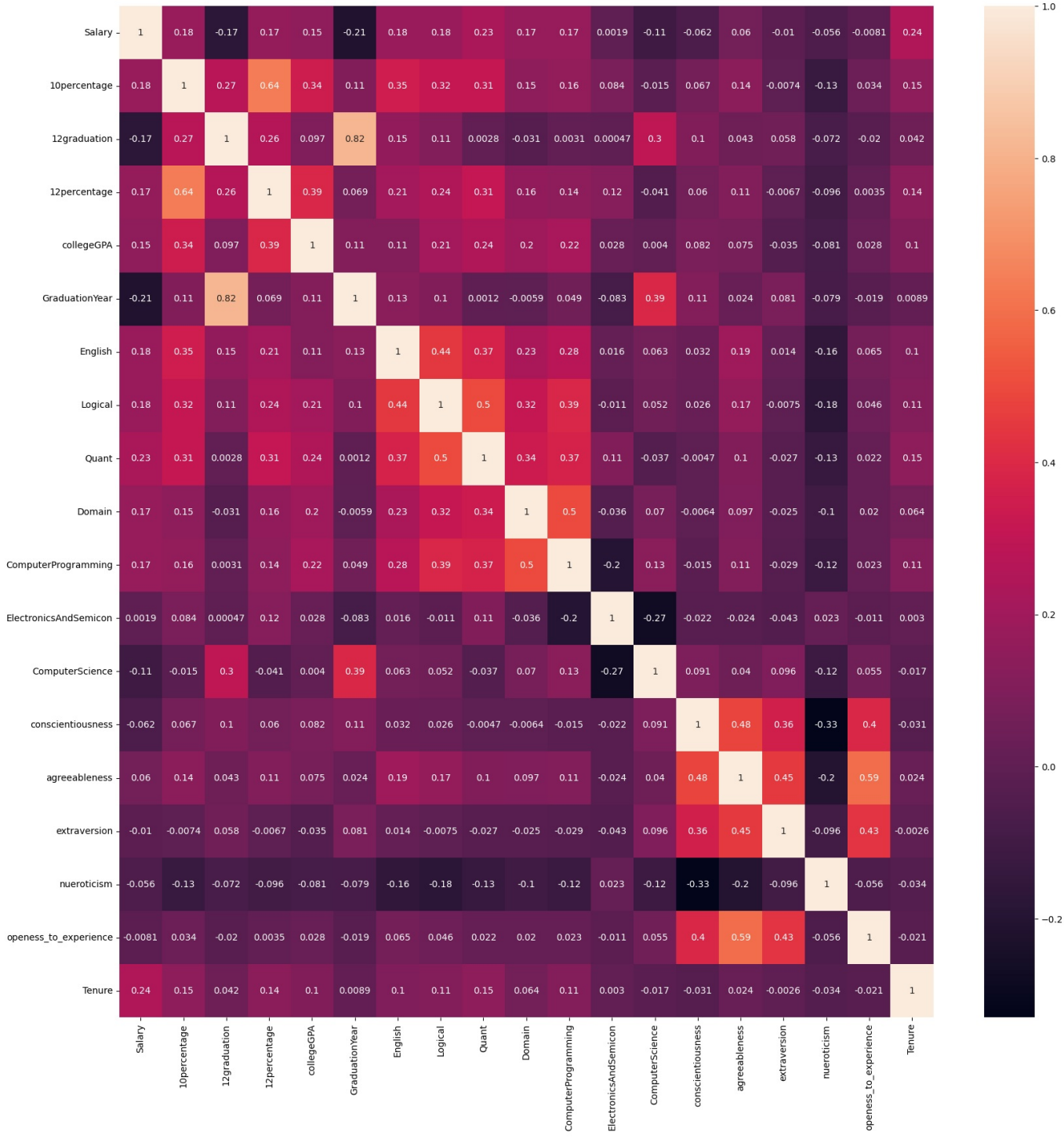


correlation plot

In [188…
```
plt.figure(figsize=(20,20))
sns.heatmap(df1.corr(),annot=True)
```

```
C:\Users\SOUMEN MONDAL\AppData\Local\Temp\ipykernel_6392\2970136069.py:2: FutureWarning: The default value of nu
meric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid col
umns or specify the value of numeric_only to silence this warning.
  sns.heatmap(df1.corr(),annot=True)
```

Out[188…
```
<AxesSubplot: >
```


Salary 1 0.18 -0.17 0.17 0.15 -0.21 0.18 0.18 0.23 0.17 0.17 0.0019 -0.11 -0.062 0.06 -0.01 -0.056 -0.0081 0.24
10percentage 0.18 1 0.27 0.64 0.34 0.11 0.35 0.32 0.31 0.15 0.16 0.084 -0.015 0.067 0.14 -0.0074 -0.13 0.034 0.15
12graduation -0.17 0.27 1 0.26 0.097 0.82 0.15 0.11 0.0028 -0.031 0.0031 0.00047 0.3 0.1 0.043 0.058 -0.072 -0.02 0.042
12percentage 0.17 0.64 0.26 1 0.39 0.069 0.21 0.24 0.31 0.16 0.14 0.12 -0.041 0.06 0.11 -0.0067 -0.096 0.0035 0.14
collegeGPA 0.15 0.34 0.097 0.39 1 0.11 0.11 0.21 0.24 0.2 0.22 0.028 0.004 0.082 0.075 -0.035 -0.081 0.028 0.1
GraduationYear -0.21 0.11 0.82 0.069 0.11 1 0.13 0.1 0.0012 -0.0059 0.049 -0.083 0.39 0.11 0.024 0.081 -0.079 -0.019 0.0089
English 0.18 0.35 0.15 0.21 0.11 0.13 1 0.44 0.37 0.23 0.28 0.016 0.063 0.032 0.19 0.014 -0.16 0.065 0.1
Logical 0.18 0.32 0.11 0.24 0.21 0.1 0.44 1 0.5 0.32 0.39 -0.011 0.052 0.026 0.17 -0.0075 -0.18 0.046 0.11
Quant 0.23 0.31 0.0028 0.31 0.24 0.0012 0.37 0.5 1 0.34 0.37 0.11 -0.037 -0.0047 0.1 -0.027 -0.13 0.022 0.15
Domain 0.17 0.15 -0.031 0.16 0.2 -0.0059 0.23 0.32 0.34 1 0.5 -0.036 0.07 -0.0064 0.097 -0.025 -0.1 0.02 0.064
ComputerProgramming 0.17 0.16 0.0031 0.14 0.22 0.049 0.28 0.39 0.37 0.5 1 -0.2 0.13 -0.015 0.11 -0.029 -0.12 0.023 0.11
ElectronicsAndSemicon 0.0019 0.084 0.00047 0.12 0.028 -0.083 0.016 -0.011 0.11 -0.036 -0.2 1 -0.27 -0.022 -0.024 -0.043 0.023 -0.011 0.003
ComputerScience -0.11 -0.015 0.3 -0.041 0.004 0.39 0.063 0.052 -0.037 0.07 0.13 -0.27 1 0.091 0.04 0.096 -0.12 0.055 -0.017
conscientiousness -0.062 0.067 0.1 0.06 0.082 0.11 0.032 0.026 -0.0047 -0.0064 -0.015 -0.022 0.091 1 0.48 0.36 -0.33 0.4 -0.031
agreeableness 0.06 0.14 0.043 0.11 0.075 0.024 0.19 0.17 0.1 0.097 0.11 -0.024 0.04 0.48 1 0.45 -0.2 0.59 0.024
extraversion -0.01 -0.0074 0.058 -0.0067 -0.035 0.081 0.014 -0.0075 -0.027 -0.025 -0.029 -0.043 0.096 0.36 0.45 1 -0.096 0.43 -0.0026
nueroticism -0.056 -0.13 -0.072 -0.096 -0.081 -0.079 -0.16 -0.18 -0.13 -0.1 -0.12 0.023 -0.12 -0.33 -0.2 -0.096 1 -0.056 -0.034
openess_to_experience -0.0081 0.034 -0.02 0.0035 0.028 -0.019 0.065 0.046 0.022 0.02 0.023 -0.011 0.055 0.4 0.59 0.43 -0.056 1 -0.021
Tenure 0.24 0.15 0.042 0.14 0.1 0.0089 0.1 0.11 0.15 0.064 0.11 0.003 -0.017 -0.031 0.024 -0.0026 -0.034 -0.021 1
Salary 10percentage 12graduation 12percentage collegeGPA GraduationYear English Logical Quant Domain ComputerProgramming ElectronicsAndSemicon ComputerScience conscientiousness agreeableness extraversion nueroticism openess_to_experience Tenure
1.0 0.8 0.6 0.4 0.2 0.0 −0.2

In [ ]:
```
sns.pairplot(data = df1, diag_kind='kde')
```

Research Questions

1. Times of India article dated Jan 18, 2019 states that “After doing your Computer Science Engineering

if you take up jobs as a Programming Analyst, Software Engineer, Hardware Engineer and Associate Engineer you can earn up to 2.5-3 lakhs as a fresh graduate.” Test this claim with the data given to you.

In [192…
```
df1['Specialization'].unique()
```

Out[192…
```
['computer engineering', 'electronics and communication engineering', 'information technology', 'computer scien
ce & engineering', 'electronics and electrical engineering', ..., 'information & communication technology', 'in
formation science', 'internal combustion engine', 'biomedical engineering', 'computer science']
Length: 42
Categories (42, object): ['aeronautical engineering', 'applied electronics and instrumentation', 'automobile/au
tomotive engineering', 'biomedical engineering', ..., 'metallurgical engineering', 'other', 'polymer technology
', 'telecommunication engineering']
```

In [193…
```
df1['Designations'].unique()
```

Out[193…
```
array(['others', 'manager', 'engineer', 'senior', 'analyst', 'associate',
       'developer', 'executive', 'trainee', 'consultant', 'designer'],
      dtype=object)
```

In [198…
```
df1['DOL'] = pd.to_datetime(df1['DOL'])
df1['DOJ'] = pd.to_datetime(df1['DOJ'])
df1['Experience']=((df1['DOL']-df1['DOJ']).map(lambda x: round(x.days/365,1)))
```

In [205…
```
df2 = df1[['Designations','Specialization','Salary','Experience']]
df2.shape
```

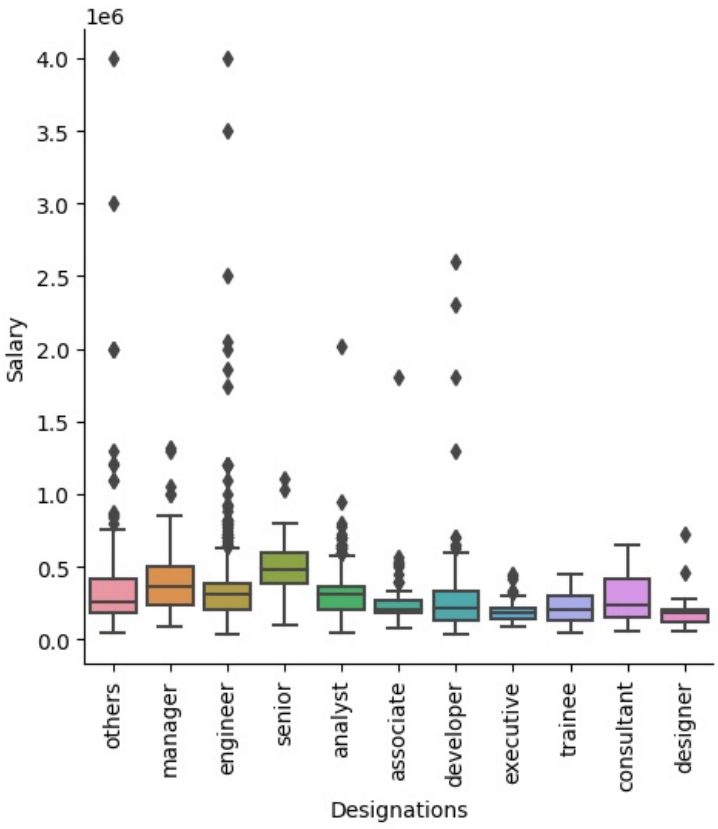
Out[205…
```
(3864, 4)
```

In [206…
```
df2.nunique()
```

Out[206…
```
Designations       11
Specialization     42
Salary            174
Experience        106
dtype: int64
```

In [207…
```
 plt.figure(figsize=(50, 50))
sns.catplot(x='Designations', y="Salary", kind="box", data=df1)
plt.xticks(rotation=90);
```

```
<Figure size 5000x5000 with 0 Axes>
```



In [208…
```
df3 = df2.loc[df1['Specialization'] == 'computer engineering']
```

In [209…
```
list=['engineer','analyst','associate']
df2=df1.loc[df1['Designations'].isin(list)]
```

In [211…
```
df3.isnull()
```

Out[211…

	Designations	Specialization	Salary	Experience
0	False	False	False	False
3	False	False	False	False
10	False	False	False	False
19	False	False	False	False
20	False	False	False	False
...	...	...	...	...
3968	False	False	False	False
3979	False	False	False	False
3984	False	False	False	False
3985	False	False	False	False
3995	False	False	False	False

582 rows × 4 columns

In [212…
```
df3.dropna(subset = ["Experience"], inplace=True)
```

In [213…
```
df3['Experience'].unique()
```

Out[213…
```
array([11.7, 12.7,  4. ,  9.5,  1.8,  2.8, 12.2, 10.4, 11. ,  2.9,  1. ,
        1.3,  3.5,  0.5,  0.2, 11.4, 10.9, 12.3,  1.4,  2.5, 11.1,  1.5,
       10.6,  2. , 11.9, 11.5,  0.7,  3.3,  0.6, 11.6, 10.1,  3.4,  1.9,
       13. , 11.2, 12.5, 11.8,  1.7,  1.1,  2.3,  9.1, 13.2, 10.7, 11.3,
       13.7, 13.6,  3. ,  9.6, 12.4, 10.8,  4.2,  2.4,  1.2,  2.2, 12. ,
        9.7,  3.8,  2.1,  4.4,  9. ,  0.3,  2.7,  0.9,  0.8,  3.2,  3.1,
        2.6,  4.9,  0.4, 10.2,  3.9, 10.5,  4.7, 13.5,  4.8,  0.1, 12.6,
       13.3, 10. , 12.9,  1.6, 12.8,  9.9, 10.3,  3.6,  3.7,  4.1,  4.5])
```

For convienince let fresher's experience taken has less than 1.0 years

In [215…
```
df4 = df3.loc[df3['Experience'] <= 1.0]
df4
```

Out[215…

	Designations	Specialization	Salary	Experience
47	developer	computer engineering	95000.0	1.0
112	others	computer engineering	170000.0	0.5
117	developer	computer engineering	180000.0	0.2
256	developer	computer engineering	95000.0	0.7
281	developer	computer engineering	110000.0	1.0
...	...	...	...	...
3867	developer	computer engineering	90000.0	0.4
3899	others	computer engineering	180000.0	0.9
3922	developer	computer engineering	240000.0	0.6
3927	engineer	computer engineering	180000.0	0.7
3979	engineer	computer engineering	550000.0	0.8

78 rows × 4 columns

- From the above information we can infer that the fresher who has done computer specialization and opted
- engineer(s/w or h/w) or analyst or associate have an average salary of 2,96,600

With maximum salary of 5.6 lakh to minimum salary of 50k

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

In [217…
```
from scipy.stats import chi2
from scipy.stats import chi2_contingency
```

In [218…
```
df1['Gender'].value_counts()
```

Out[218…
```
m    2932
f     932
Name: Gender, dtype: int64
```

In [219…
```
df1['Specialization'].value_counts()
```

Out[219…
```
electronics and communication engineering      856
computer science & engineering                 714
information technology                         649
computer engineering                           582
computer application                           232
mechanical engineering                         194
electronics and electrical engineering         185
electronics & telecommunications               119
electrical engineering                          79
electronics & instrumentation eng               32
civil engineering                               28
information science engineering                 27
electronics and instrumentation engineering     26
instrumentation and control engineering         19
electronics engineering                         18
biotechnology                                   15
other                                           11
applied electronics and instrumentation          9
industrial & production engineering              9
chemical engineering                             8
telecommunication engineering                    6
automobile/automotive engineering                5
computer science and technology                  5
instrumentation engineering                      4
mechatronics                                     4
mechanical and automation                        4
electronics and computer engineering             3
aeronautical engineering                         3
information & communication technology           2
industrial engineering                           2
biomedical engineering                           2
metallurgical engineering                        2
information science                              1
computer science                                 1
computer and communication engineering           1
electrical and power engineering                 1
internal combustion engine                       1
mechanical & production engineering              1
industrial & management engineering              1
control and instrumentation engineering          1
polymer technology                               1
ceramic engineering                              1
Name: Specialization, dtype: int64
```

In [220…
```
pd.crosstab(df['Specialization'], df['Gender'], margins=True)
```

Out[220…

Gender	f	m	All
Specialization			
aeronautical engineering	1	2	3
applied electronics and instrumentation	2	7	9
automobile/automotive engineering	0	5	5
biomedical engineering	2	0	2
biotechnology	9	6	15
ceramic engineering	0	1	1
chemical engineering	1	8	9
civil engineering	6	23	29
computer and communication engineering	0	1	1
computer application	58	183	241
computer engineering	173	419	592
computer networking	0	1	1
computer science	1	1	2
computer science & engineering	181	558	739
computer science and technology	2	4	6
control and instrumentation engineering	0	1	1
electrical and power engineering	0	2	2
electrical engineering	16	65	81
electronics	0	1	1
electronics & instrumentation eng	10	21	31
electronics & telecommunications	28	93	121
electronics and communication engineering	211	664	875
electronics and computer engineering	0	3	3
electronics and electrical engineering	34	160	194
electronics and instrumentation engineering	5	21	26
electronics engineering	3	16	19
embedded systems technology	0	1	1
industrial & management engineering	0	1	1
industrial & production engineering	2	8	10
industrial engineering	1	1	2
information & communication technology	2	0	2
information science	0	1	1
information science engineering	8	19	27
information technology	173	482	655
instrumentation and control engineering	9	10	19
instrumentation engineering	0	4	4
internal combustion engine	0	1	1
mechanical & production engineering	0	1	1
mechanical and automation	0	5	5
mechanical engineering	10	187	197
mechatronics	1	3	4
metallurgical engineering	0	2	2
other	0	13	13
polymer technology	0	1	1
power systems and automation	0	1	1
telecommunication engineering	1	5	6
All	950	3012	3962

In [221…

```
observed = pd.crosstab(df['Specialization'], df['Gender'])
```

```
observed
```

Out[221…

Gender	f	m
Specialization		
aeronautical engineering	1	2
applied electronics and instrumentation	2	7
automobile/automotive engineering	0	5
biomedical engineering	2	0
biotechnology	9	6
ceramic engineering	0	1
chemical engineering	1	8
civil engineering	6	23
computer and communication engineering	0	1
computer application	58	183
computer engineering	173	419
computer networking	0	1
computer science	1	1
computer science & engineering	181	558
computer science and technology	2	4
control and instrumentation engineering	0	1
electrical and power engineering	0	2
electrical engineering	16	65
electronics	0	1
electronics & instrumentation eng	10	21
electronics & telecommunications	28	93
electronics and communication engineering	211	664
electronics and computer engineering	0	3
electronics and electrical engineering	34	160
electronics and instrumentation engineering	5	21
electronics engineering	3	16
embedded systems technology	0	1
industrial & management engineering	0	1
industrial & production engineering	2	8
industrial engineering	1	1
information & communication technology	2	0
information science	0	1
information science engineering	8	19
information technology	173	482
instrumentation and control engineering	9	10
instrumentation engineering	0	4
internal combustion engine	0	1
mechanical & production engineering	0	1
mechanical and automation	0	5
mechanical engineering	10	187
mechatronics	1	3
metallurgical engineering	0	2
other	0	13
polymer technology	0	1
power systems and automation	0	1
telecommunication engineering	1	5

In [222…

```
chi2_contingency(observed)
```

Out[222…
```
(104.4818911512974,
 1.240545079268252e-06,
 45,
 array([[7.19333670e-01, 2.28066633e+00],
        [2.15800101e+00, 6.84199899e+00],
        [1.19888945e+00, 3.80111055e+00],
        [4.79555780e-01, 1.52044422e+00],
        [3.59666835e+00, 1.14033317e+01],
        [2.39777890e-01, 7.60222110e-01],
        [2.15800101e+00, 6.84199899e+00],
        [6.95355881e+00, 2.20464412e+01],
        [2.39777890e-01, 7.60222110e-01],
        [5.77864715e+01, 1.83213529e+02],
        [1.41948511e+02, 4.50051489e+02],
        [2.39777890e-01, 7.60222110e-01],
        [4.79555780e-01, 1.52044422e+00],
        [1.77195861e+02, 5.61804139e+02],
        [1.43866734e+00, 4.56133266e+00],
        [2.39777890e-01, 7.60222110e-01],
        [4.79555780e-01, 1.52044422e+00],
        [1.94220091e+01, 6.15779909e+01],
        [2.39777890e-01, 7.60222110e-01],
        [7.43311459e+00, 2.35668854e+01],
        [2.90131247e+01, 9.19868753e+01],
        [2.09805654e+02, 6.65194346e+02],
        [7.19333670e-01, 2.28066633e+00],
        [4.65169107e+01, 1.47483089e+02],
        [6.23422514e+00, 1.97657749e+01],
        [4.55577991e+00, 1.44442201e+01],
        [2.39777890e-01, 7.60222110e-01],
        [2.39777890e-01, 7.60222110e-01],
        [2.39777890e+00, 7.60222110e+00],
        [4.79555780e-01, 1.52044422e+00],
        [4.79555780e-01, 1.52044422e+00],
        [2.39777890e-01, 7.60222110e-01],
        [6.47400303e+00, 2.05259970e+01],
        [1.57054518e+02, 4.97945482e+02],
        [4.55577991e+00, 1.44442201e+01],
        [9.59111560e-01, 3.04088844e+00],
        [2.39777890e-01, 7.60222110e-01],
        [2.39777890e-01, 7.60222110e-01],
        [1.19888945e+00, 3.80111055e+00],
        [4.72362443e+01, 1.49763756e+02],
        [9.59111560e-01, 3.04088844e+00],
        [4.79555780e-01, 1.52044422e+00],
        [3.11711257e+00, 9.88288743e+00],
        [2.39777890e-01, 7.60222110e-01],
        [2.39777890e-01, 7.60222110e-01],
        [1.43866734e+00, 4.56133266e+00]]))
```

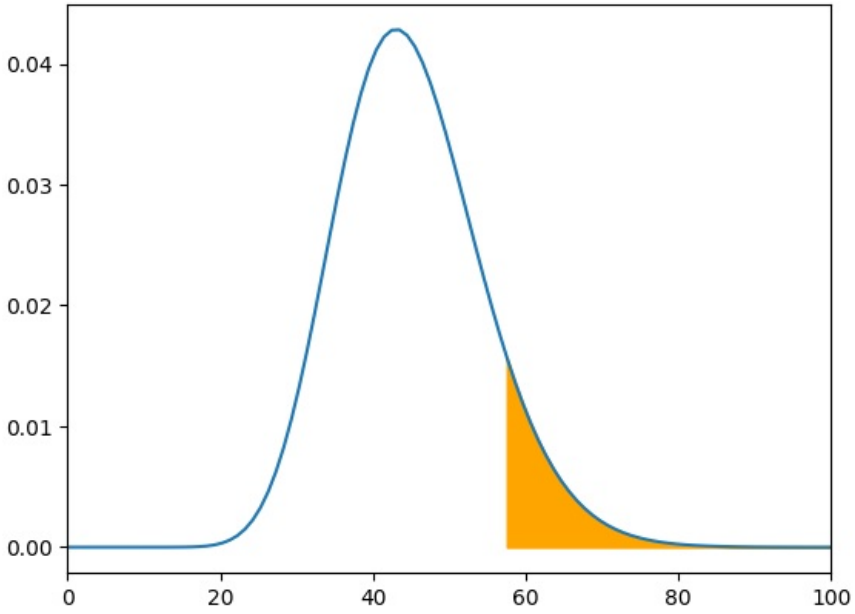
In [223…
```
# Computing chi2 test statistic, p-value, degree of freedoms
chi2_test_stat = chi2_contingency(observed)[0]
pval = chi2_contingency(observed)[1]
df1 = chi2_contingency(observed)[2]
```

In [225…
```
confidence_level = 0.90
alpha = 1 - confidence_level
chi2_critical = chi2.ppf(1 - alpha, df1)
chi2_critical
```

Out[225…
```
57.50530474499599
```

In [226…
```
# Ploting the chi2 distribution to visualise
# Defining the x minimum and x maximum
x_min = 0
x_max = 100
# Ploting the graph and setting the x limits
x = np.linspace(x_min, x_max, 100)
y = chi2.pdf(x, df1)
plt.xlim(x_min, x_max)
plt.plot(x, y)
# Setting Chi2 Critical value
chi2_critical_right = chi2_critical
# Shading the right rejection region
x1 = np.linspace(chi2_critical_right, x_max, 100)
y1 = chi2.pdf(x1, df1)
plt.fill_between(x1, y1, color='orange')
```

Out[226…
```
<matplotlib.collections.PolyCollection at 0x2361d063f10>
```



In [227…
```
if(chi2_test_stat > chi2_critical):
 print("Reject Null Hypothesis")
else:
 print("Fail to Reject Null Hypothesis")
```

```
Reject Null Hypothesis
```

In [228…
```
if(pval < alpha):
 print("Reject Null Hypothesis")
else:
 print("Fail to Reject Null Hypothesis")
```

```
Reject Null Hypothesis
```

From the above test it rejects null hypothesis and hence we can say both are dependent

- From the EDA we observe that , the specialization is dependent on the gender
- The fresher who has done CSE and have got into analyst,engineer etc position have an maximum salary of

5.6lakh and minimum of 50k

- With experience the there is a hike in the salary

In [ ]:

In [ ]: