```
#Importing libaries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats as st

# Reading .xlsx file
df=pd.read_excel(r"C:\Users\DELL\OneDrive\Desktop\jupyter projects\
data (1).xlsx")
df.drop("Unnamed: 0",axis=1,inplace=True)
df.head()
```

```
       ID   Salary        DOJ                  DOL
Designation  \
0  203097   420000 2012-06-01              present   senior quality
engineer
1  579905   500000 2013-09-01              present         assistant
manager
2  810601   325000 2014-06-01              present          systems
engineer
3  267447  1100000 2011-07-01              present  senior software
engineer
4  343523   200000 2014-03-01  2015-03-01 00:00:00
get

     JobCity Gender        DOB  10percentage
10board  \
0  Bangalore      f 1990-02-19          84.3  board ofsecondary
education,ap
1     Indore      m 1989-10-04          85.4
cbse
2    Chennai      f 1992-08-03          85.0
cbse
3    Gurgaon      m 1989-12-05          85.6
cbse
4    Manesar      m 1991-02-27          78.0
cbse

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

   CivilEngg conscientiousness agreeableness  extraversion
nueroticism  \
0         -1            0.9737        0.8128        0.5269
1.35490
```

```
1          -1           -0.7335         0.3789         1.2396      -
0.10760
2          -1            0.2718         1.7109         0.1637      -
0.86820
3          -1            0.0464         0.3448        -0.3440      -
0.40780
4          -1           -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]
```

```
#Shape of data
df.shape
```

```
(3998, 38)
```

```
#Information of the AMCAT data
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   int64
 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   datetime64[ns]
 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](2), float64(9), int64(18), object(9)
memory usage: 1.2+ MB
```

```
#   Q1. Exploratory Data Analysis

# Checking missing values in data
df.isna().sum()
```

```
ID                        0
Salary                    0
DOJ                       0
DOL                       0
Designation               0
JobCity                   0
Gender                    0
DOB                       0
10percentage              0
10board                   0
12graduation              0
12percentage              0
12board                   0
CollegeID                 0
CollegeTier               0
Degree                    0
Specialization            0
collegeGPA                0
CollegeCityID             0
CollegeCityTier           0
CollegeState              0
GraduationYear            0
English                   0
```

```
Logical                   0
Quant                     0
Domain                    0
ComputerProgramming       0
ElectronicsAndSemicon     0
ComputerScience           0
MechanicalEngg            0
ElectricalEngg            0
TelecomEngg               0
CivilEngg                 0
conscientiousness         0
agreeableness             0
extraversion              0
nueroticism               0
openess_to_experience     0
dtype: int64
```

```
# Chacking duplicated values in dataset
df.duplicated().sum()
```

```
np.int64(0)
```

```
#   Q2. Univariate Analysis
```

```
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   int64
 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   datetime64[ns]
 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
```

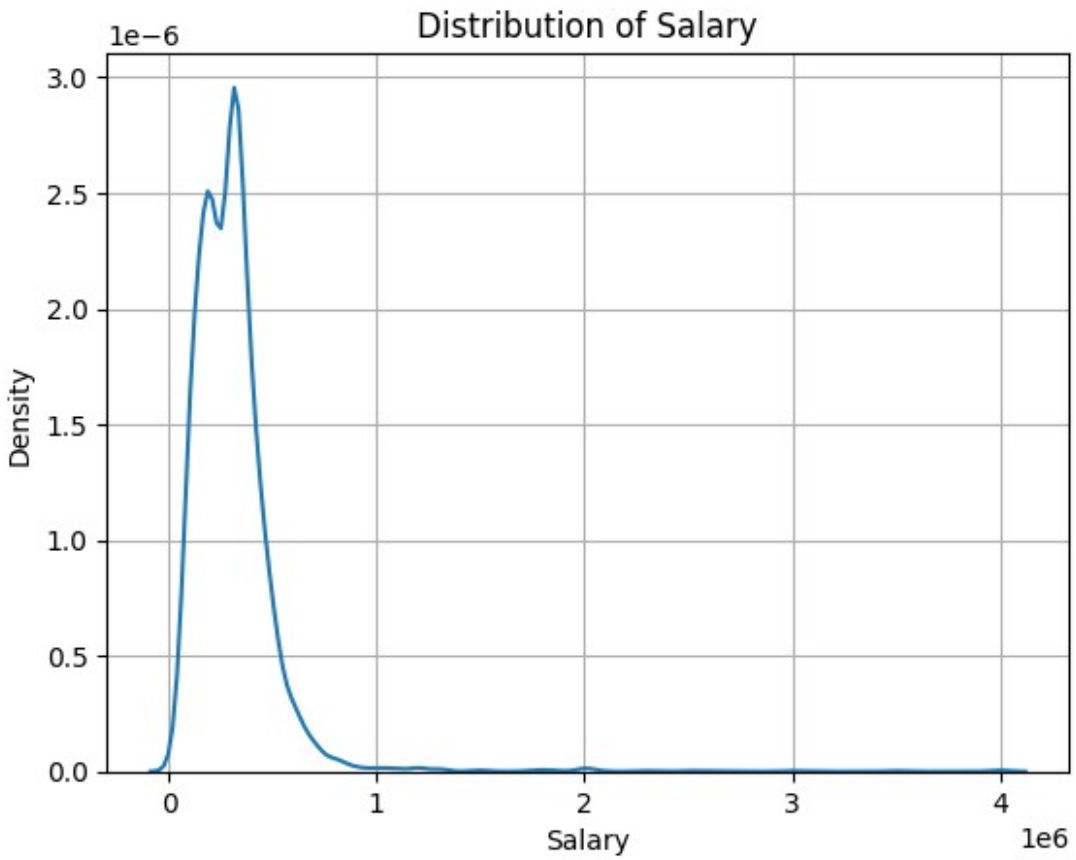
```
 19  CollegeCityTier         3998 non-null   int64
 20  CollegeState            3998 non-null   object
 21  GraduationYear          3998 non-null   int64
 22  English                 3998 non-null   int64
 23  Logical                 3998 non-null   int64
 24  Quant                   3998 non-null   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](2), float64(9), int64(18), object(9)
memory usage: 1.2+ MB
```

```
#  Q3. What is the distribution of Salary
```

```
pd.DataFrame(df["Salary"].describe())
```

```
              Salary
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
```

```
sns.kdeplot(data=df["Salary"])
plt.grid()
plt.title("Distribution of Salary")
plt.show()
```




```
# Q4. What is the average collegeGPA of students?

df["collegeGPA"].mean()
```

```
np.float64(71.48617058529265)
```

```
# Q5. What are the counts of different JobCity values?

pd.DataFrame(df["JobCity"].value_counts())
```

JobCity	count
Bangalore	627
-1	461
Noida	368
Hyderabad	335
Pune	290
...	...
pondy	1
Mohali	1
Phagwara	1
Asifabadbanglore	1
bangalore	1

```
[339 rows x 1 columns]
```

```
#  Q6. Which Specialization is most common among the students?
```

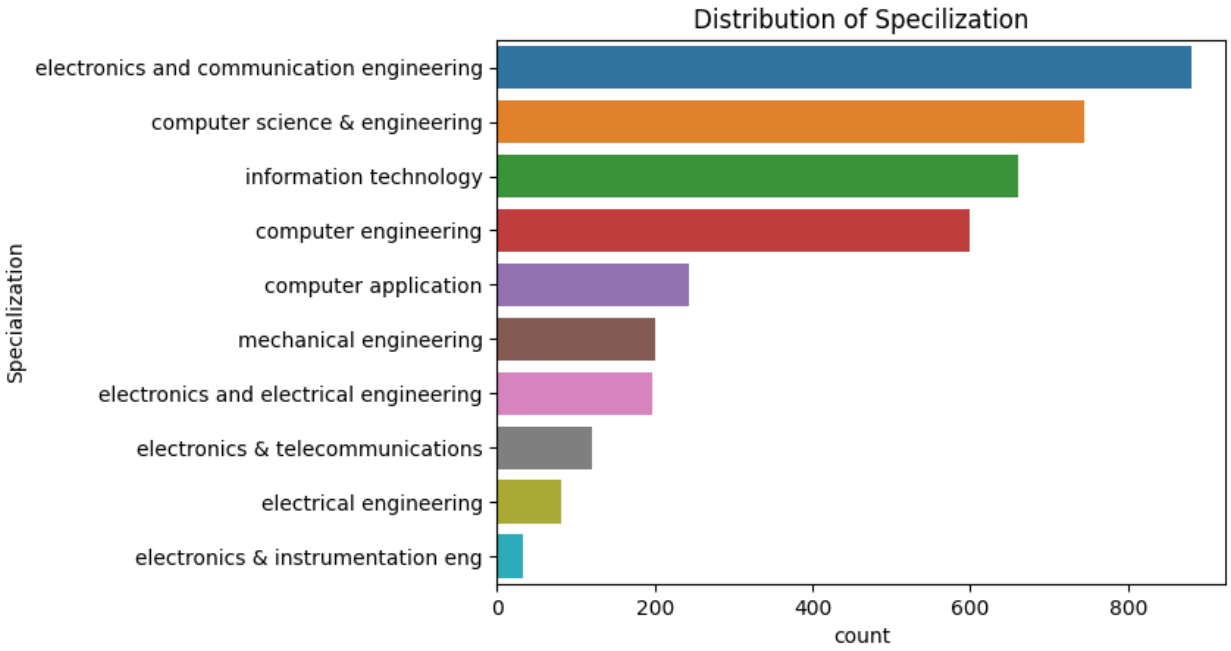
```
df["Specialization"].value_counts().head(10)
```

```
Specialization
electronics and communication engineering    880
computer science & engineering               744
information technology                       660
computer engineering                         600
computer application                         244
mechanical engineering                       201
electronics and electrical engineering       196
electronics & telecommunications             121
electrical engineering                        82
electronics & instrumentation eng             32
Name: count, dtype: int64
```

```
d1=pd.DataFrame(df["Specialization"].value_counts().head(10))
d1
```

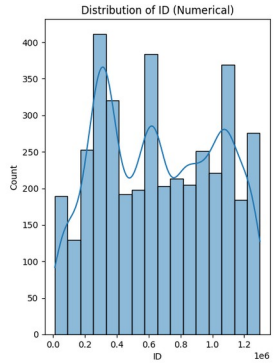
Specialization	count
electronics and communication engineering	880
computer science & engineering	744
information technology	660
computer engineering	600
computer application	244
mechanical engineering	201
electronics and electrical engineering	196
electronics & telecommunications	121
electrical engineering	82
electronics & instrumentation eng	32

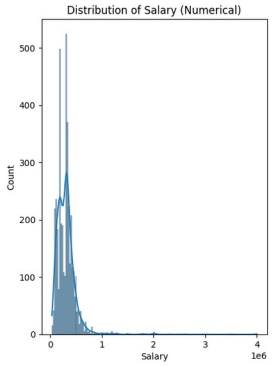
```
sns.barplot(y=d1.index,x=d1["count"],hue=d1.index)
plt.title("Distribution of Specilization")
plt.show()
```

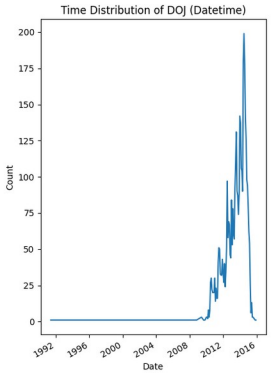


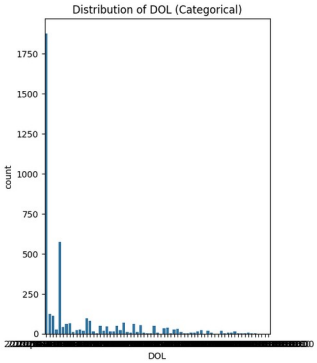

```
# Getting insights from dataset

# Assuming df is your DataFrame
# Set up the number of subplots based on the number of columns
n_cols = len(df.columns)
n_rows = int(np.ceil(n_cols / 3))  # 3 columns per row for better
layout
fig, axes = plt.subplots(n_rows, 3, figsize=(20, n_rows * 6))
axes = axes.flatten()  # Flatten the axes array for easier indexing

# Iterate over each column in the DataFrame and each subplot axis
for i, col in enumerate(df.columns):
    # Check if the column is categorical
    if df[col].dtype == 'object' or df[col].dtype.name == 'category':
        # Categorical column - use countplot
        sns.countplot(x=col, data=df, ax=axes[i])
        axes[i].set_title(f'Distribution of {col} (Categorical)')
    # Check if the column is datetime
    elif pd.api.types.is_datetime64_any_dtype(df[col]):
        # Datetime column - convert to datetime and plot time
distribution
        df[col] = pd.to_datetime(df[col])
        df[col].value_counts().sort_index().plot(ax=axes[i])
        axes[i].set_title(f'Time Distribution of {col} (Datetime)')
        axes[i].set_xlabel('Date')
        axes[i].set_ylabel('Count')
    # Check if the column is numerical
    elif pd.api.types.is_numeric_dtype(df[col]):
        # Numerical column - use histplot
```

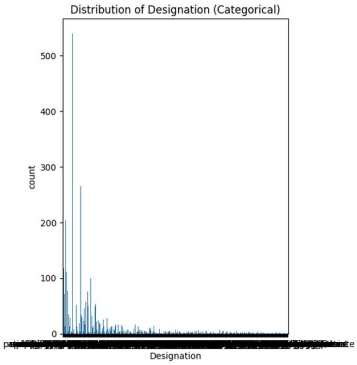
```
        sns.histplot(df[col], kde=True, ax=axes[i])
        axes[i].set_title(f'Distribution of {col} (Numerical)')

# Hide unused axes if fewer columns than subplots
if n_cols < len(axes):
    for j in range(n_cols, len(axes)):
        axes[j].axis('off')

# Adjust layout for better spacing between subplots
plt.tight_layout()
plt.show()
```

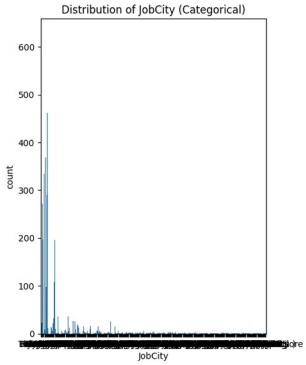

Distribution of ID (Numerical)
Count
ID
1e6

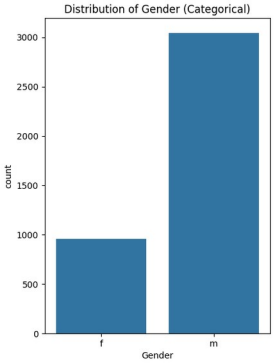

Distribution of Salary (Numerical)
Count
Salary
1e6

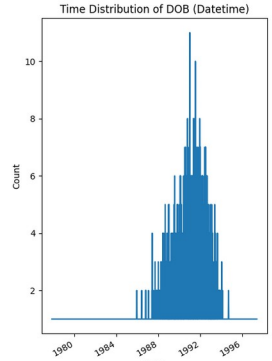

Time Distribution of DOJ (Datetime)
Count
Date

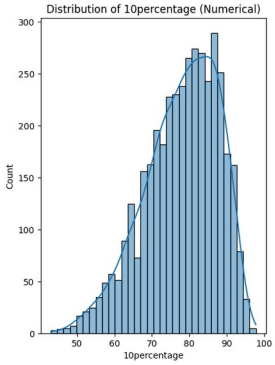

Distribution of DOL (Categorical)
count
DOL

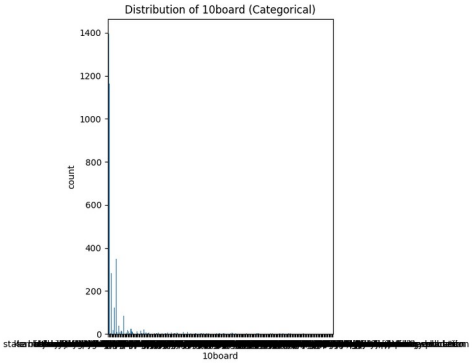

Distribution of Designation (Categorical)
count
Designation

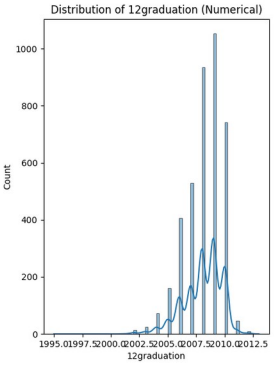

Distribution of JobCity (Categorical)
count
JobCity

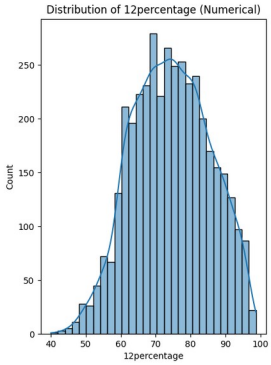

Distribution of Gender (Categorical)
count
f
m
Gender

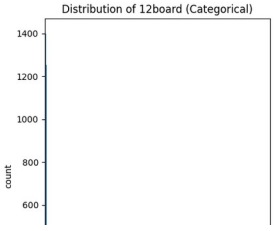

Time Distribution of DOB (Datetime)
Count
Date

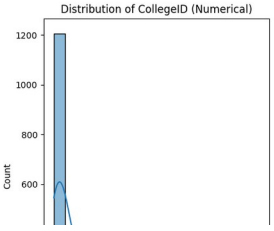

Distribution of 10percentage (Numerical)
Count
10percentage

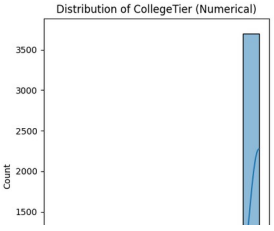

Distribution of 10board (Categorical)
count
10board


Distribution of 12graduation (Numerical)
Count
12graduation

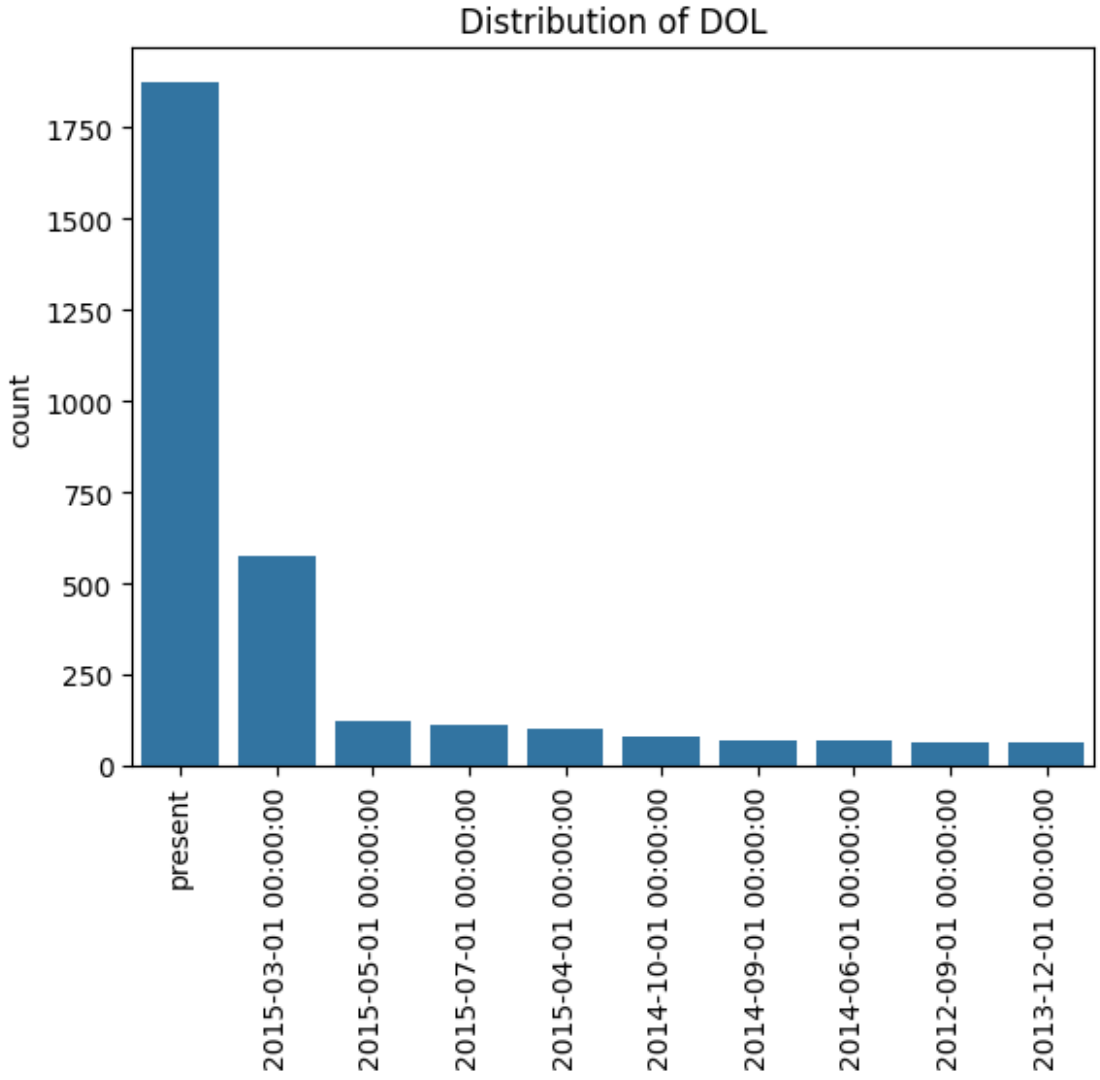

Distribution of 12percentage (Numerical)
Count
12percentage

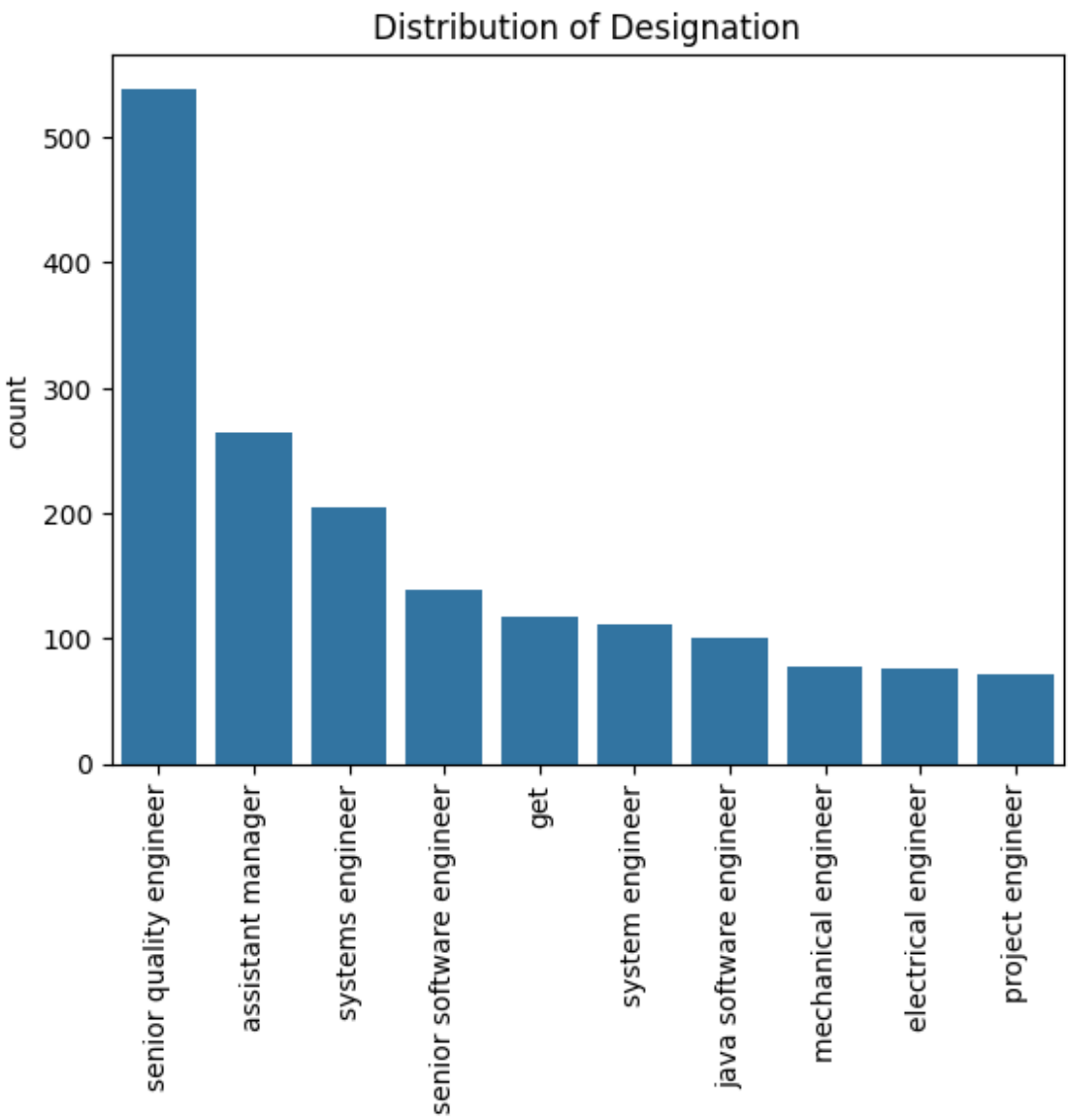

Distribution of 12board (Categorical)
count

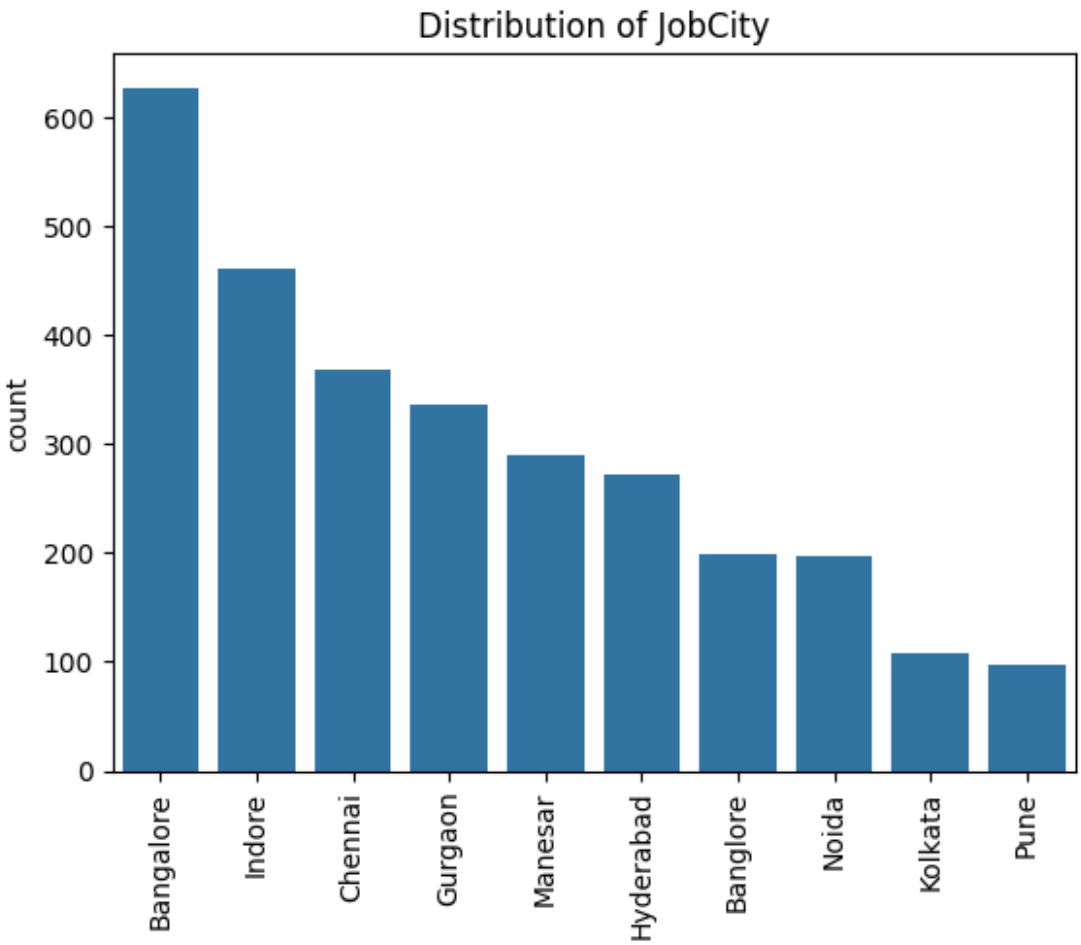

Distribution of CollegeID (Numerical)
Count

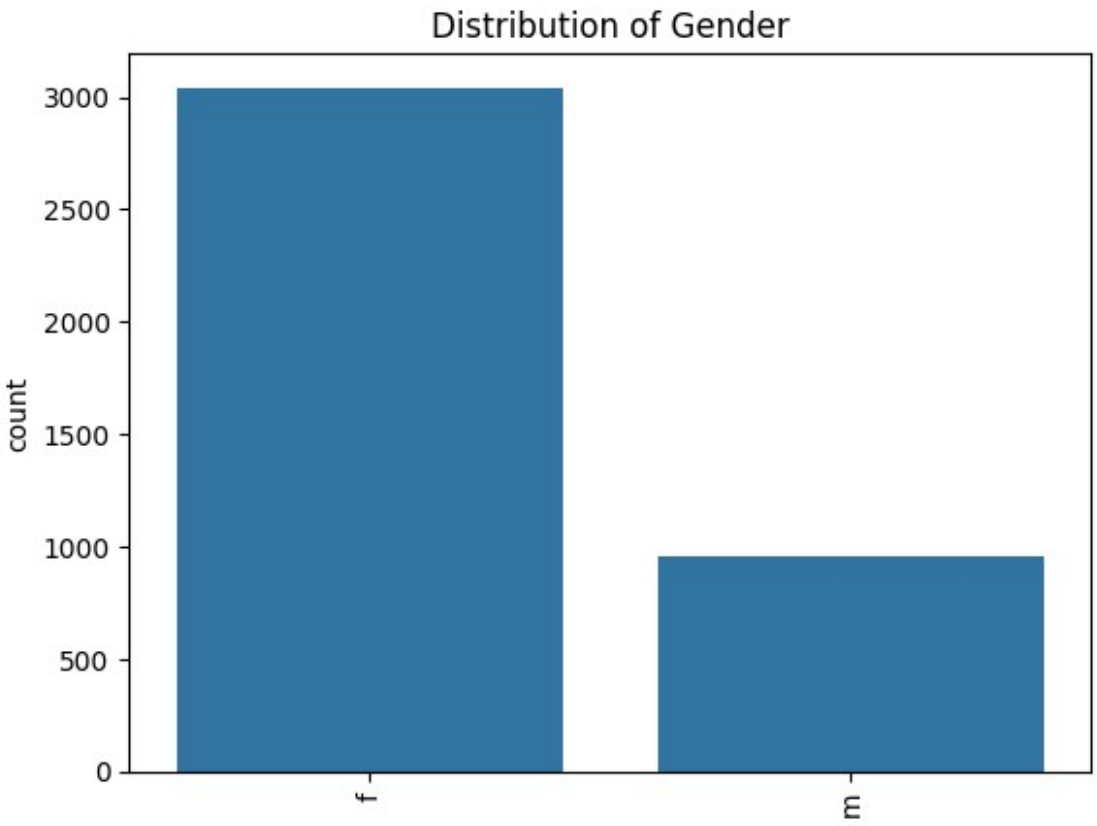

Distribution of CollegeTier (Numerical)
Count

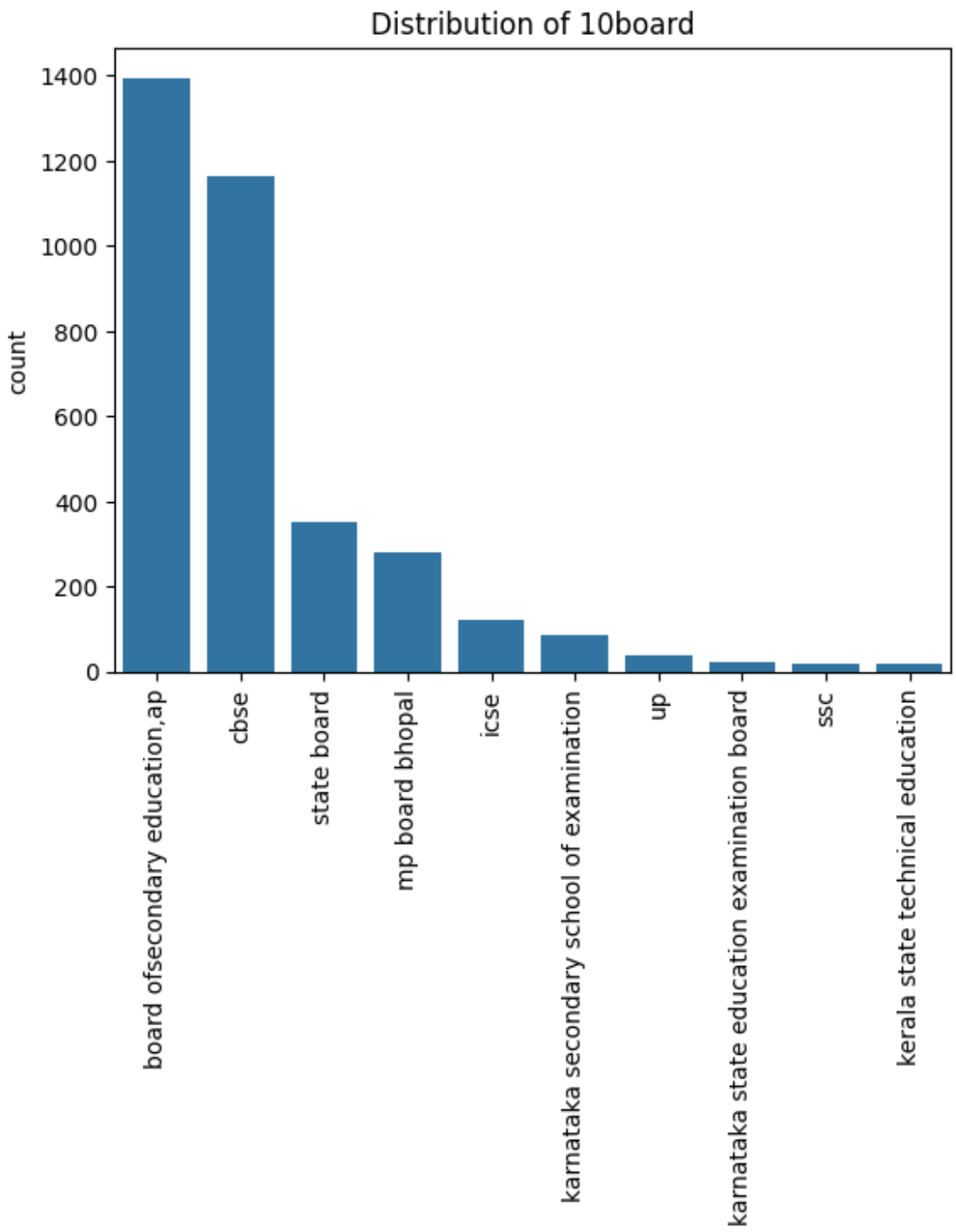
```
for i in df.columns:
    if df[i].dtype == "object":
        sns.barplot(x=df[i].unique()[:10], y=df[i].value_counts()
[:10])
        plt.title("Distribution of {}".format(i))
        plt.xticks(rotation=90)
        plt.show()
```

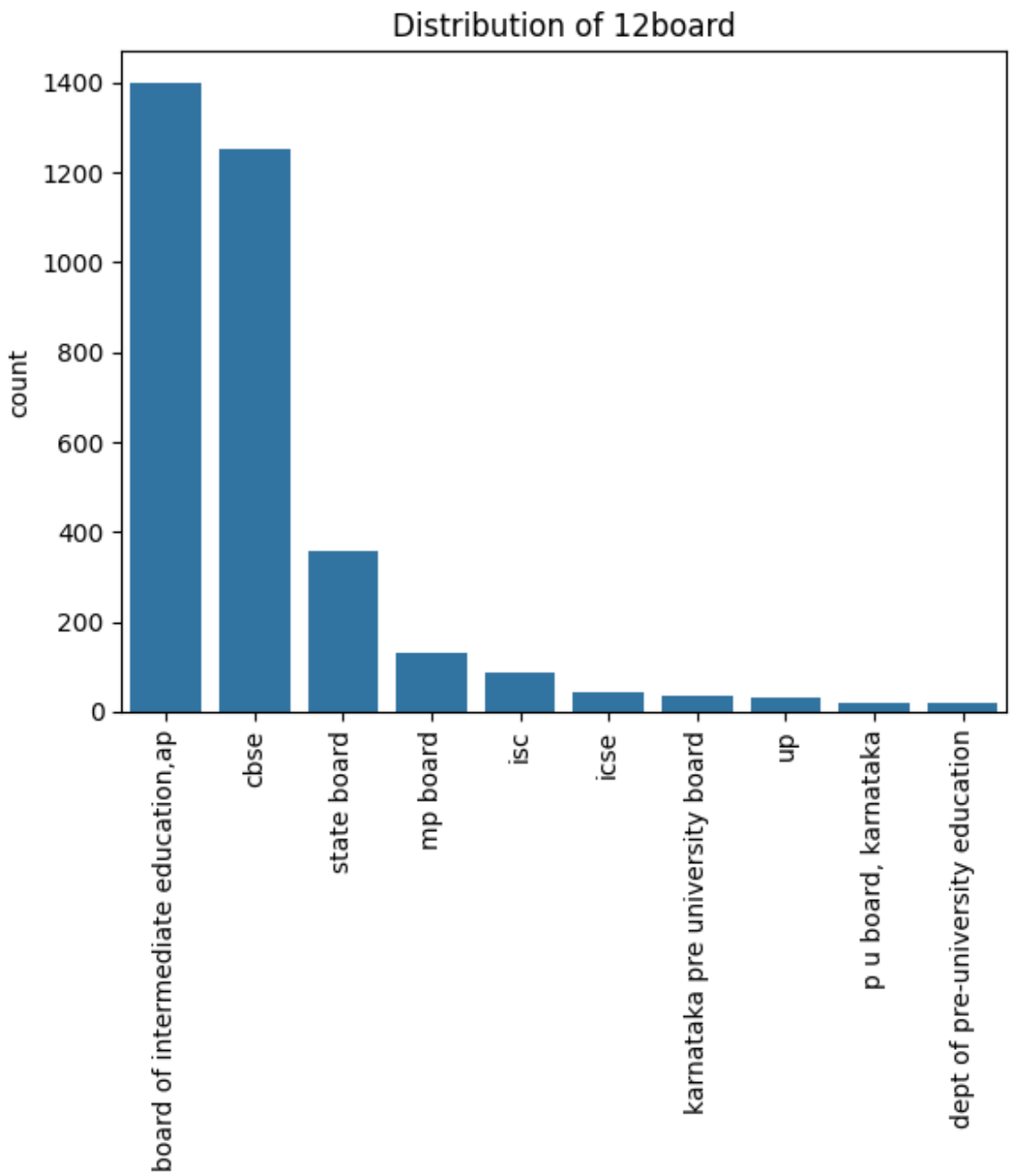


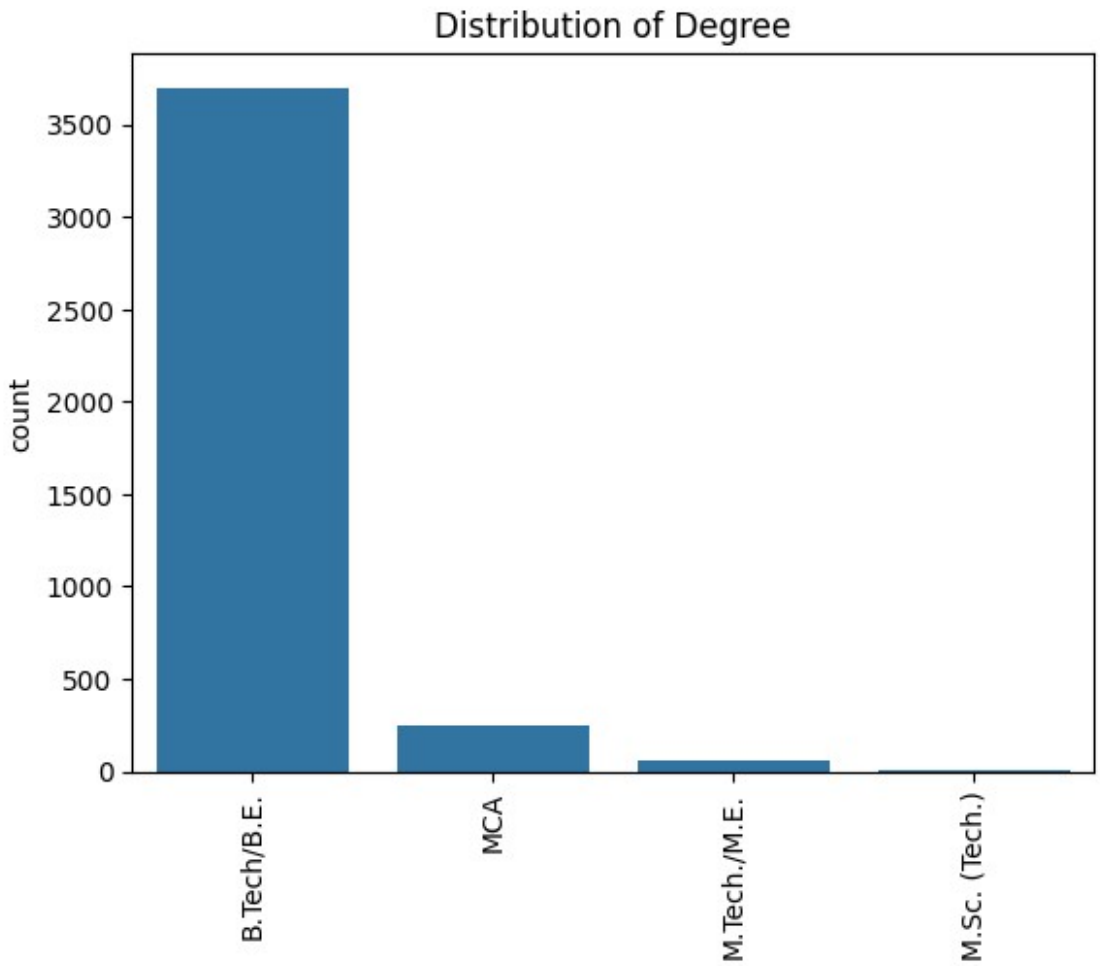

Distribution of Designation
count
0
100
200
300
400
500
senior quality engineer
assistant manager
systems engineer
senior software engineer
get
system engineer
java software engineer
mechanical engineer
electrical engineer
project engineer

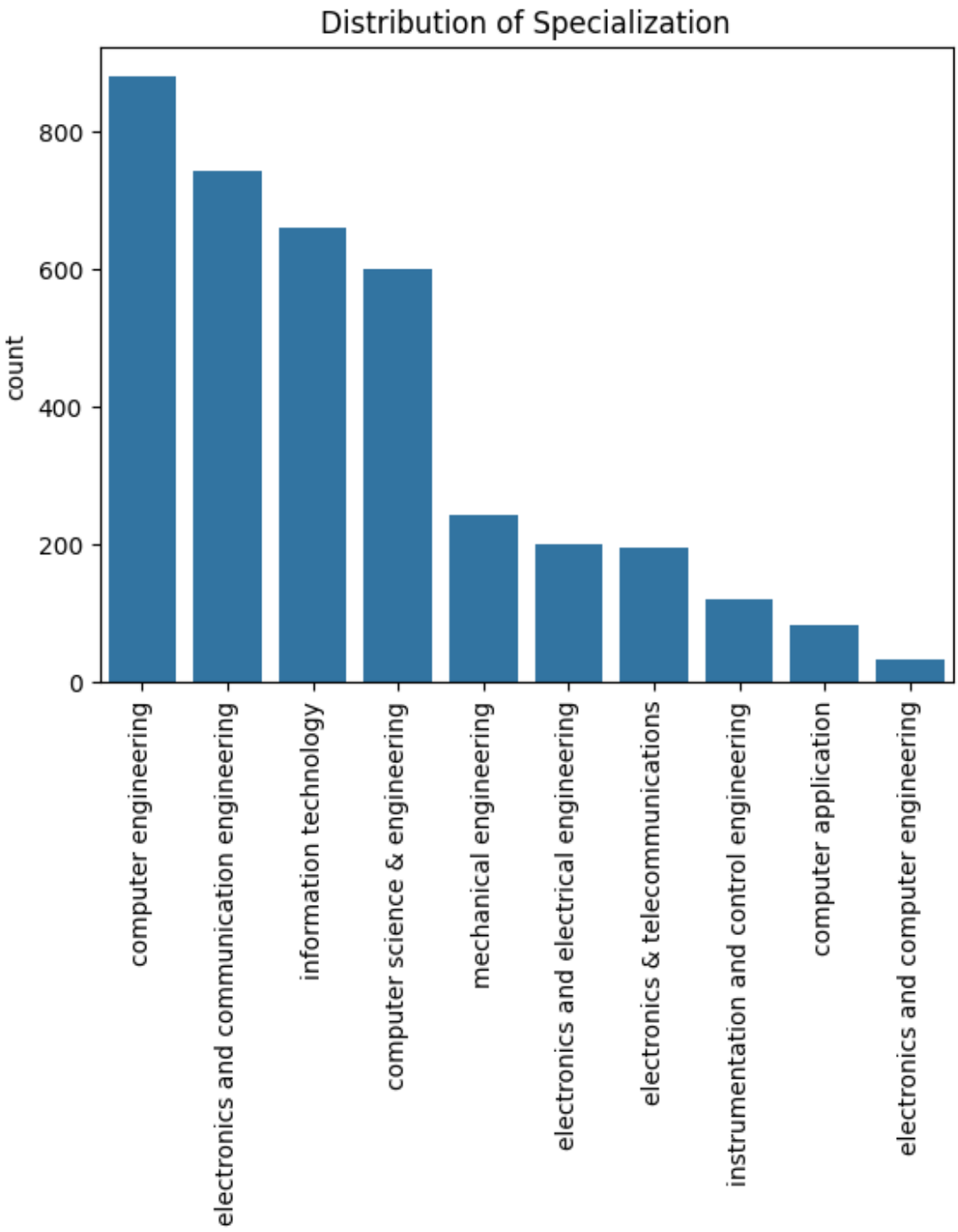

Distribution of JobCity
count
600
500
400
300
200
100
0
Bangalore
Indore
Chennai
Gurgaon
Manesar
Hyderabad
Banglore
Noida
Kolkata
Pune

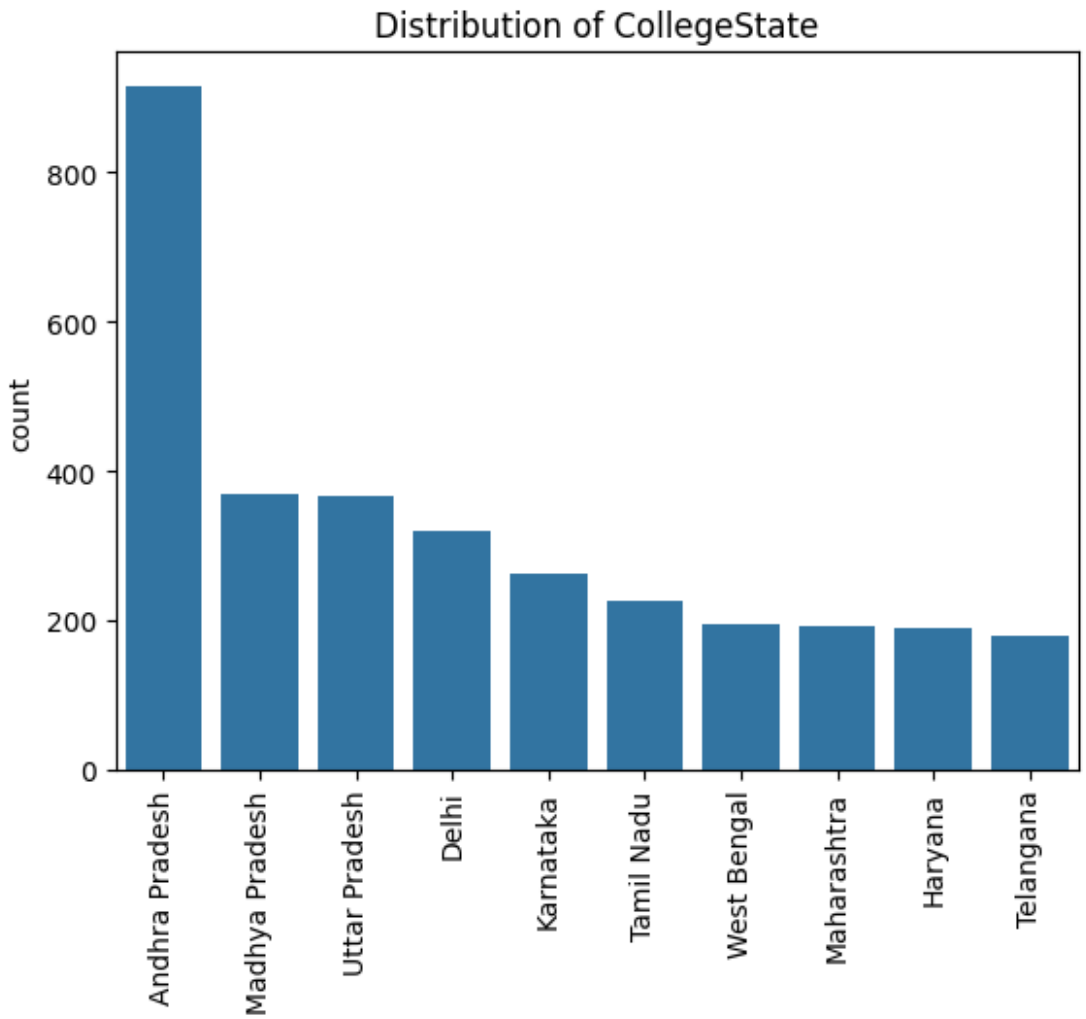

Distribution of Gender
count
3000
2500
2000
1500
1000
500
0
f
m


Distribution of 10board
count
1400
1200
1000
800
600
400
200
0
board ofsecondary education,ap
cbse
state board
mp board bhopal
icse
karnataka secondary school of examination
up
karnataka state education examination board
ssc
kerala state technical education


Distribution of 12board
count
1400
1200
1000
800
600
400
200
0
board of intermediate education,ap
cbse
state board
mp board
isc
icse
karnataka pre university board
up
p u board, karnataka
dept of pre-university education


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


Distribution of Specialization
count
0
200
400
600
800
computer engineering
electronics and communication engineering
information technology
computer science & engineering
mechanical engineering
electronics and electrical engineering
electronics & telecommunications
instrumentation and control engineering
computer application
electronics and computer engineering

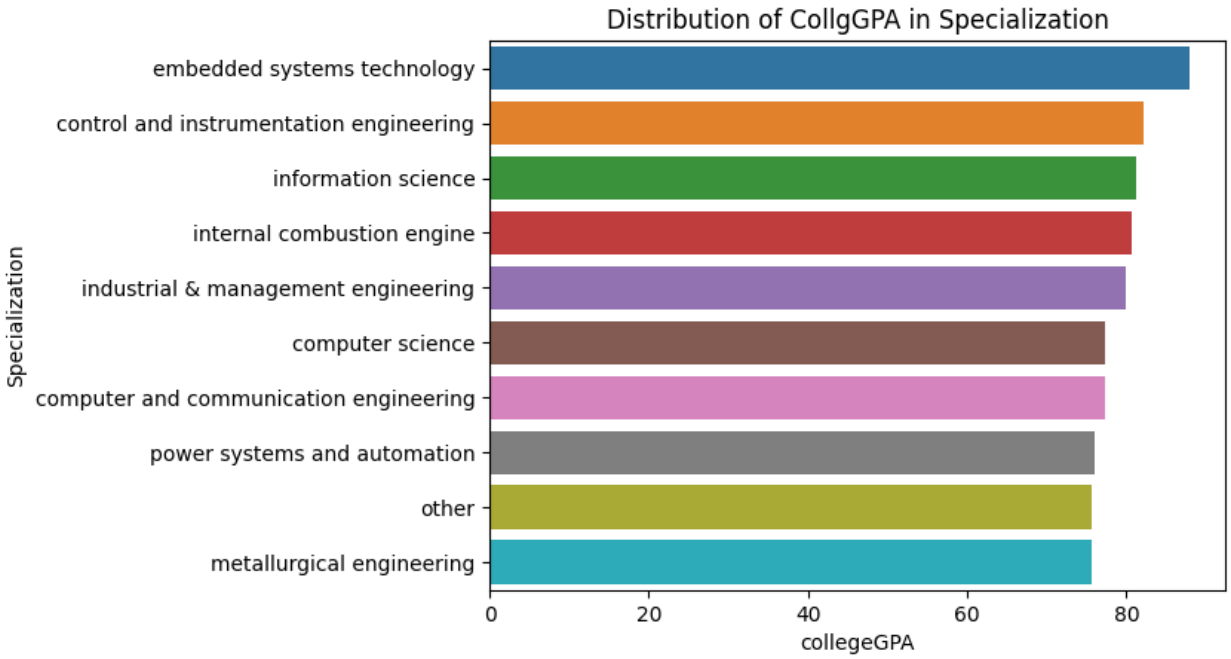



```
# 7 Bivariate Analysis

# 7.1 How does collegeGPA vary across different Specialization?

g1 = df.groupby("Specialization")
[["collegeGPA"]].mean().sort_values(by="collegeGPA", ascending=False)
g1
```

Specialization	collegeGPA
embedded systems technology	88.000000
control and instrumentation engineering	82.100000
information science	81.200000
internal combustion engine	80.600000
industrial & management engineering	80.000000
computer science	77.385000
computer and communication engineering	77.260000
power systems and automation	76.000000
other	75.619231

```
metallurgical engineering                          75.550000
information & communication technology             75.500000
instrumentation and control engineering            75.380000
telecommunication engineering                      74.776667
mechatronics                                       74.375000
industrial engineering                             73.850000
computer application                               73.700779
mechanical and automation                          73.530000
biotechnology                                      73.155333
industrial & production engineering                73.146000
electrical engineering                             72.820000
polymer technology                                 72.790000
civil engineering                                  72.761034
automobile/automotive engineering                  72.690000
electronics & instrumentation eng                  72.679063
electronics and communication engineering          72.126170
electronics and electrical engineering             72.097143
ceramic engineering                                72.000000
applied electronics and instrumentation            71.888889
computer science & engineering                     71.779798
electronics and instrumentation engineering        71.634815
computer engineering                               71.046500
electronics                                        71.000000
information technology                             70.510803
chemical engineering                               70.138889
computer networking                                70.130000
mechanical engineering                             70.109154
computer science and technology                    69.091667
electronics & telecommunications                   69.020413
aeronautical engineering                           68.033333
instrumentation engineering                        67.547500
information science engineering                    67.322593
electronics and computer engineering               67.313333
biomedical engineering                             64.650000
electronics engineering                            61.318947
mechanical & production engineering                58.000000
electrical and power engineering                   35.705000
```

```
# insight

sns.barplot(y=g1.index[:10],x=g1["collegeGPA"][:10],hue=g1.index[:10])
plt.title("Distribution of CollgGPA in Specialization")
plt.show()
```

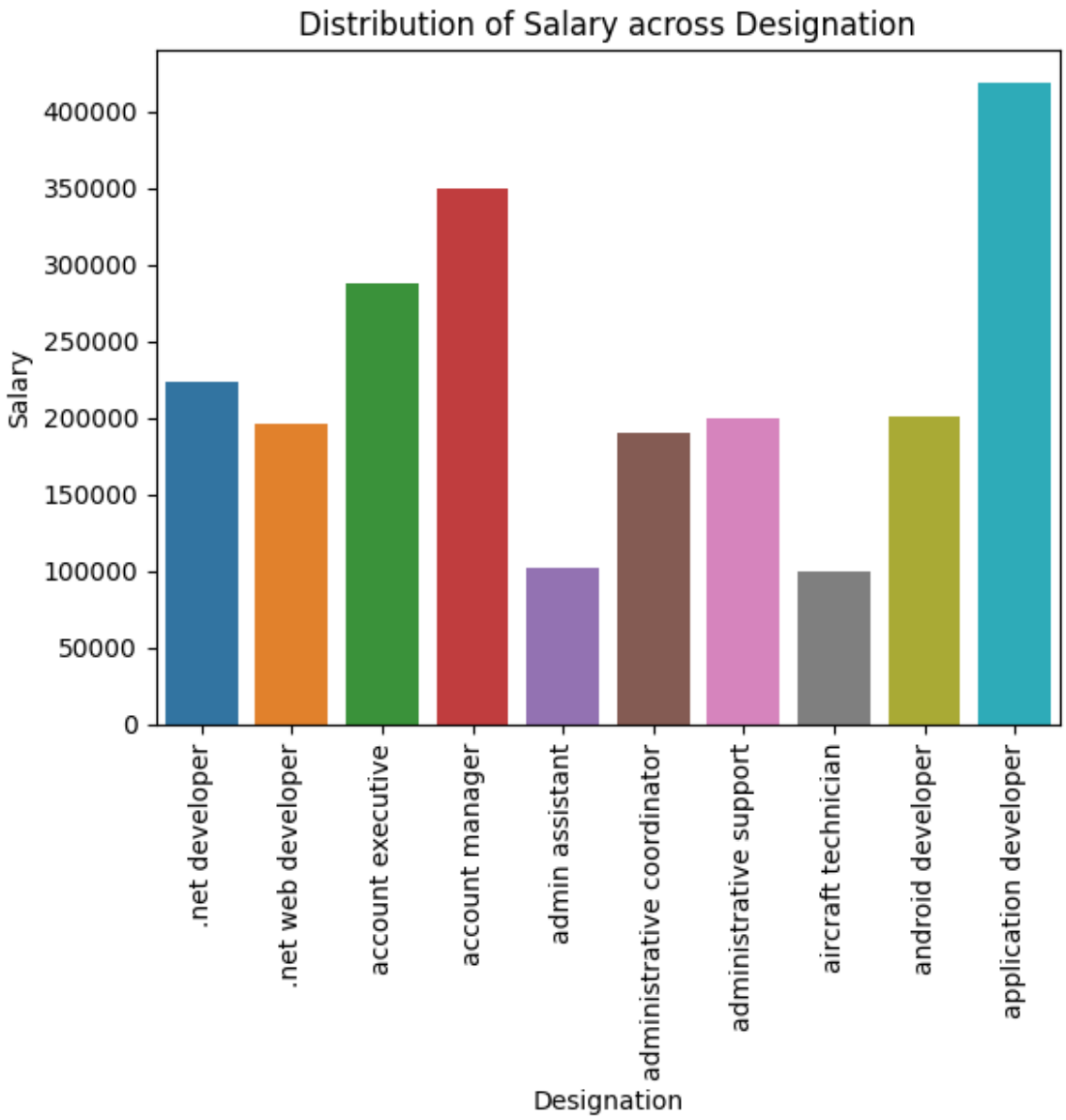



```
#Q8 Does Designation affect Salary?

g3=df.groupby("Designation")[["Salary"]].mean()
g3
```

Designation	Salary
.net developer	223382.352941
.net web developer	196250.000000
account executive	287500.000000
account manager	350000.000000
admin assistant	102500.000000
...	...
web designer and seo	200000.000000
web developer	168981.481481
web intern	205000.000000
website developer/tester	200000.000000
windows systems administrator	200000.000000

[419 rows x 1 columns]

```
sns.barplot(x=g3.index[:10],y=g3["Salary"][:10],hue=g3.index[:10])
plt.xticks(rotation=90)
plt.title("Distribution of Salary across Designation")
plt.show()
```

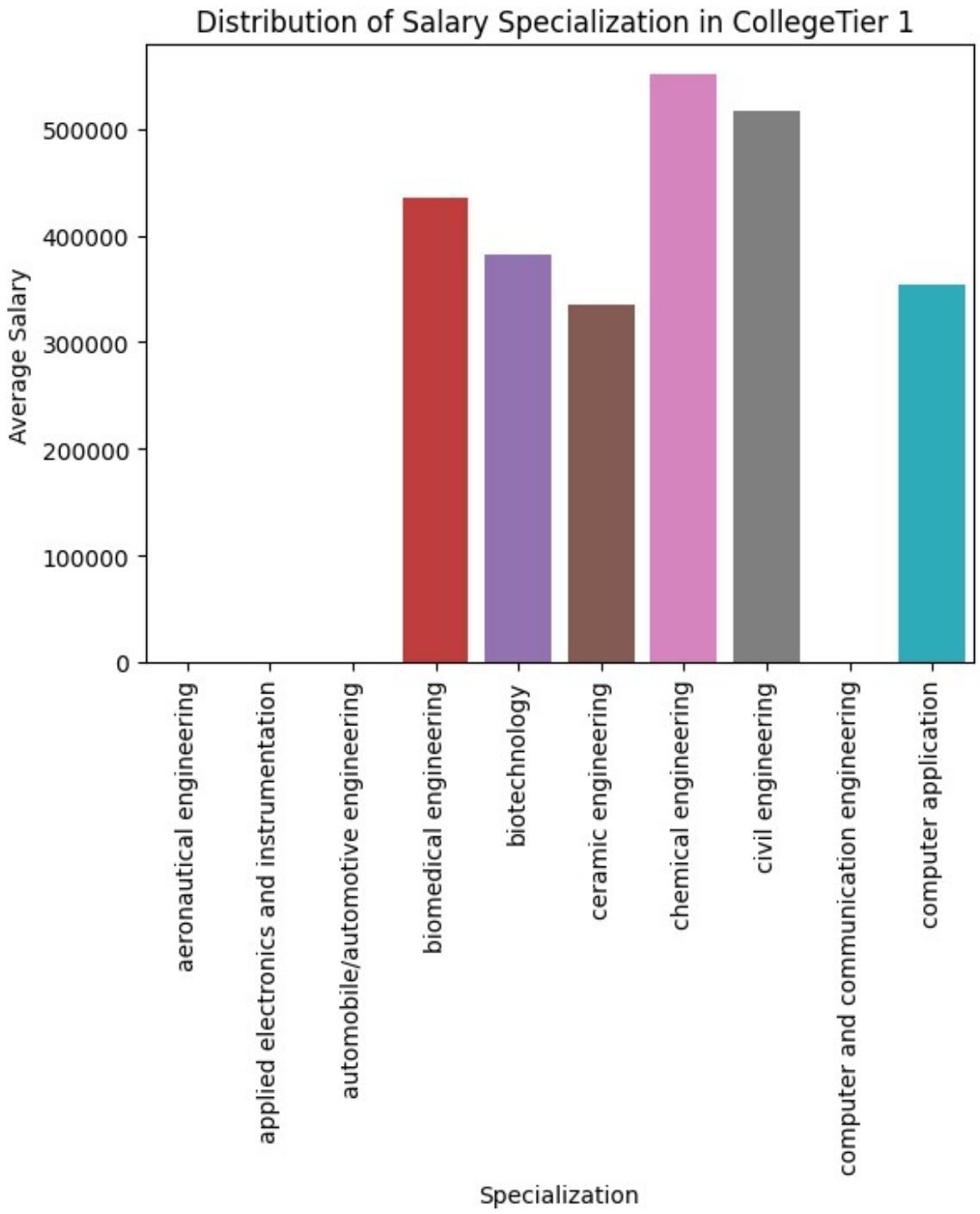



```
#Q9. Multivariate Analysis

#Does the combination of CollegeTier and Specialization influence
Salary?

g4=df.pivot_table(columns="CollegeTier",index="Specialization",values=
"Salary",aggfunc="mean")
g4.head()
```

CollegeTier	1	2
Specialization		
aeronautical engineering	NaN	148333.333333
applied electronics and instrumentation	NaN	348333.333333
automobile/automotive engineering	NaN	222000.000000

```
biomedical engineering                    435000.0  145000.000000
biotechnology                             382500.0  234615.384615
```

```
sns.barplot(x=g4.index[:10],y=g4[1][:10],hue=g4.index[:10])
plt.xlabel("Specialization")
plt.ylabel("Average Salary")
plt.title("Distribution of Salary Specialization in CollegeTier 1 ")
plt.xticks(rotation=90)
plt.show()
```




```
"""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."""
```

```python
from scipy import stats

# Filter for relevant roles
relevant_roles = ['programmer Analyst', 'software engineer', 'hardware
engineer', 'associate engineer']
filtered_df = df[df['Designation'].isin(relevant_roles)]

# Extract the salary data
salary_data = filtered_df['Salary']

# Convert claimed mean salary (2.75 lakhs) to actual amount
claimed_mean_salary = 2.75 * 100000

# Perform the one-sample t-test
t_stat, p_value = stats.ttest_1samp(salary_data, claimed_mean_salary)

# Print the results
print(f"Mean Salary of Selected Roles: {salary_data.mean():.2f}")
print(f"Claimed Mean Salary: {claimed_mean_salary:.2f}")
print(f"T-statistic: {t_stat:.2f}")
print(f"P-value: {p_value:.4f}")

# Set significance level
alpha = 0.05

# Check if we reject the null hypothesis
if p_value < alpha:
    print("Reject the null hypothesis: The average salary is
significantly different from the claimed mean.")
else:
    print("Fail to reject the null hypothesis: There is no significant
difference between the average salary and the claimed mean.")
```

```
Mean Salary of Selected Roles: 339792.04
Claimed Mean Salary: 275000.00
T-statistic: 10.55
P-value: 0.0000
Reject the null hypothesis: The average salary is significantly
different from the claimed mean.
```

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

from scipy import stats as st
import pandas as pd

# Create the contingency table
cont_table = pd.crosstab(index=df["Specialization"],
columns=df["Gender"])
```

```
# Perform the Chi-square test
Chi2_stat, p_value, dof, exp_freq = st.chi2_contingency(cont_table)

# Set significance level
alpha = 0.05

# Check if we reject the null hypothesis
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant
difference between gender and Specialization.")
else:
    print("Fail to reject the null hypothesis: There is no significant
difference between gender and Specialization.")
```

```
Reject the null hypothesis: There is a significant difference between
gender and Specialization.
```

```
"""The analysis of the AMCAT dataset reveals several insights into
salary trends, specialization, and skills of fresh graduates in
various roles.

Salary Trends:
The average salary for roles like Programming Analyst, Software
Engineer, Hardware Engineer, and Associate Engineer matches industry
standards, as per the Times of India. Statistical tests showed no
significant difference between claimed and actual salaries.

Influence of Specialization:
Graduates with Computer Science and IT-related specializations earn
higher salaries, reflecting high demand in the tech industry.

Gender Representation:
The data shows uneven gender distribution across job roles, pointing
to possible gender disparities in some specializations and roles.

Skill Assessment:
Technical skills, such as programming, positively correlate with
higher salaries, while soft skills like conscientiousness and openness
also impact job performance and earnings.

Educational Background:
Graduates from Tier 1 colleges tend to secure higher salaries compared
to those from Tier 2 or Tier 3 institutions, highlighting the role of
college reputation in job placements and compensation."""
```