INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

## PROJECT ON

# EDA Project - AMCAT Data Analysis

*by: Rayapudi Gautam Kumar*

# About me

- My name is Gautam, I’ve pursued my bachelor’s in EEE(Electrical and Electronics Engineering) and am a recent graduate and am driven towards tech-industry.
- Since I am from an electrical background, I am good in Math and with my interest in coding I thought data field would be perfect for a guy like me.
- I don’t have any prior work experience as of now and am looking for oppurtunities where I can best implement my skills. Internships like the one Innomatics is providing are an immense help to the people who are seeking to upskill their career and knowledge
- My LinkedIn Profile ID: *https://www.linkedin.com/in/gautamrayapudi*
- My Github Profile ID: *https://github.com/GautamRayapudi*

# Agenda

## Business Problem

- AMCAT (Aspiring Minds Computer Adaptive Test) is an employability assessment test used by companies to evaluate the job-readiness of candidates. The test assesses various skills such as aptitude, technical knowledge, and communication skills. Here we test the employability and various factors effecting the recruitment.

## Objective of the Project

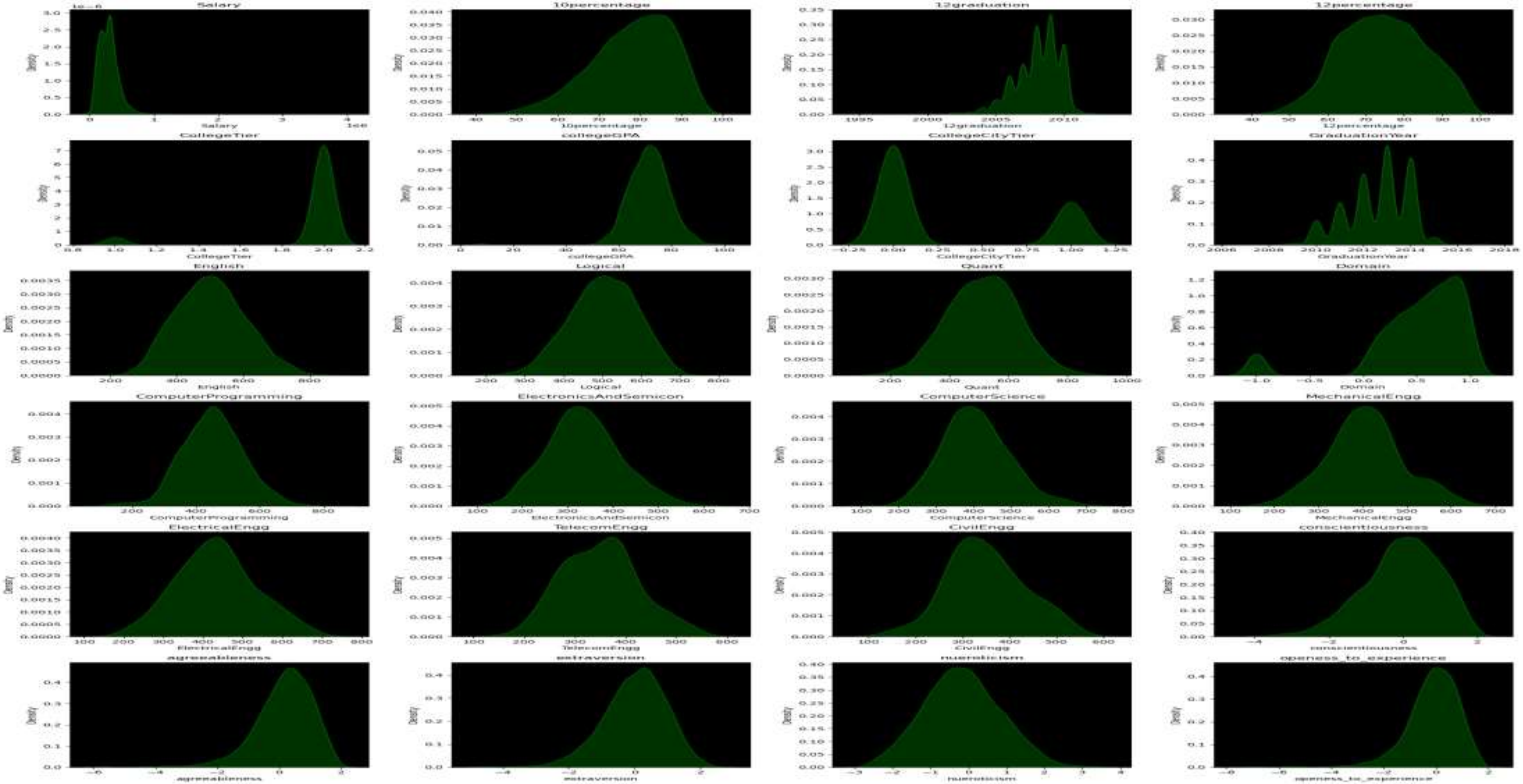
- The main objective of this project is to assess the various factors affecting the employability of the candidates through AMCAT exam 2015

# Exploratory Data Analysis:

- *Data Cleaning Steps:*
- Removed the outliers selectively based on the effect it had on the analysis.
- Looked for duplicates and null values if available.
- Corrected the datatypes of the columns and typecasted if needed.
- Extracted the values with correct spelling and no repetition using fuzzywuzzy.
- *Data Manipulation Steps:*
- Created new columns to make our analysis more accurate and expansive.
- While plotting, I dropped columns which had no relevance or dependency with the analysis like ID,CollegeID, and CollegeCityID.
- I needed to merge some dataframes to get some of the plots in a personalized way.

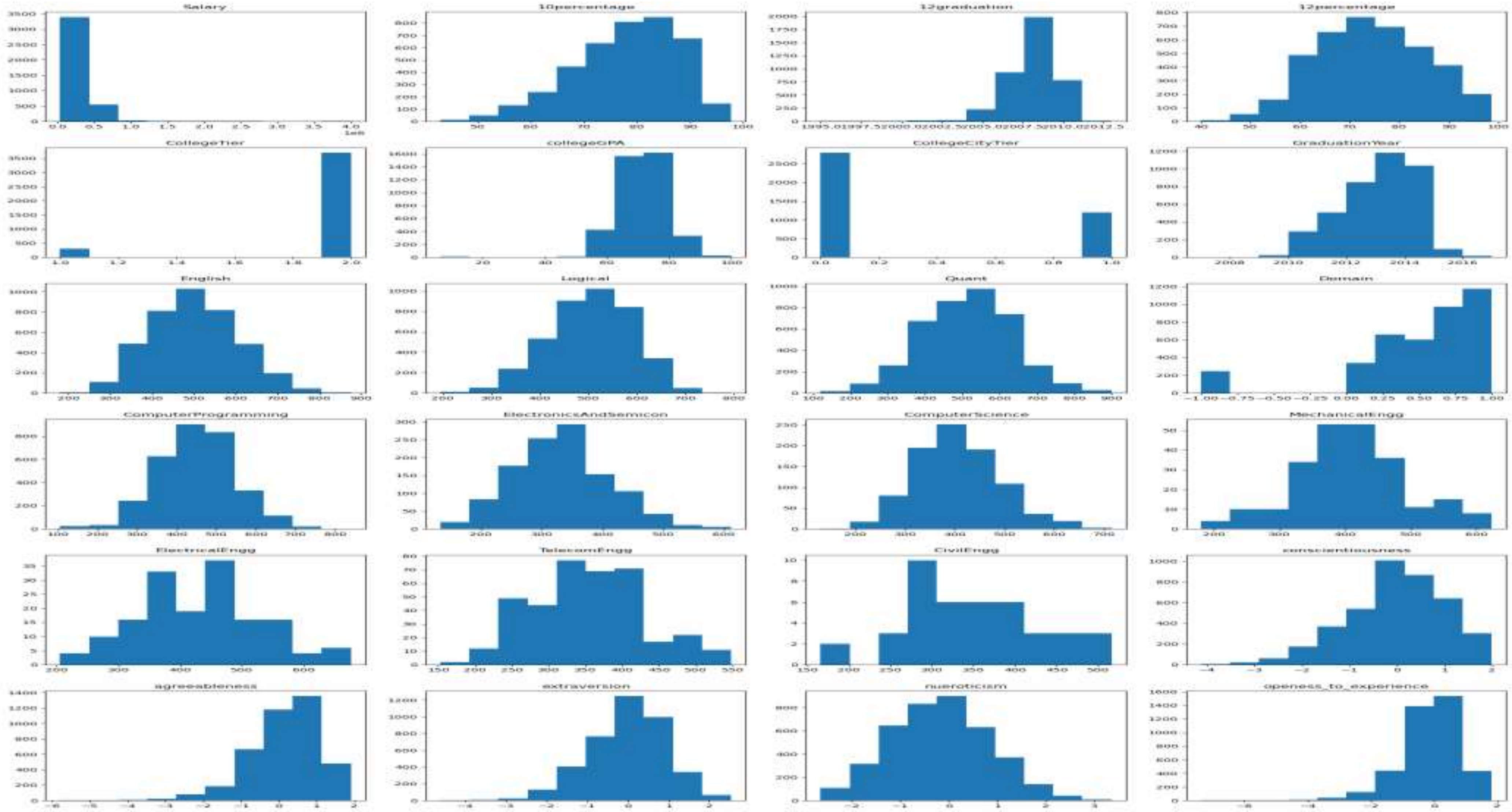
# Univariate Analysis  Steps

KDE Plot(For Numerical Columns)

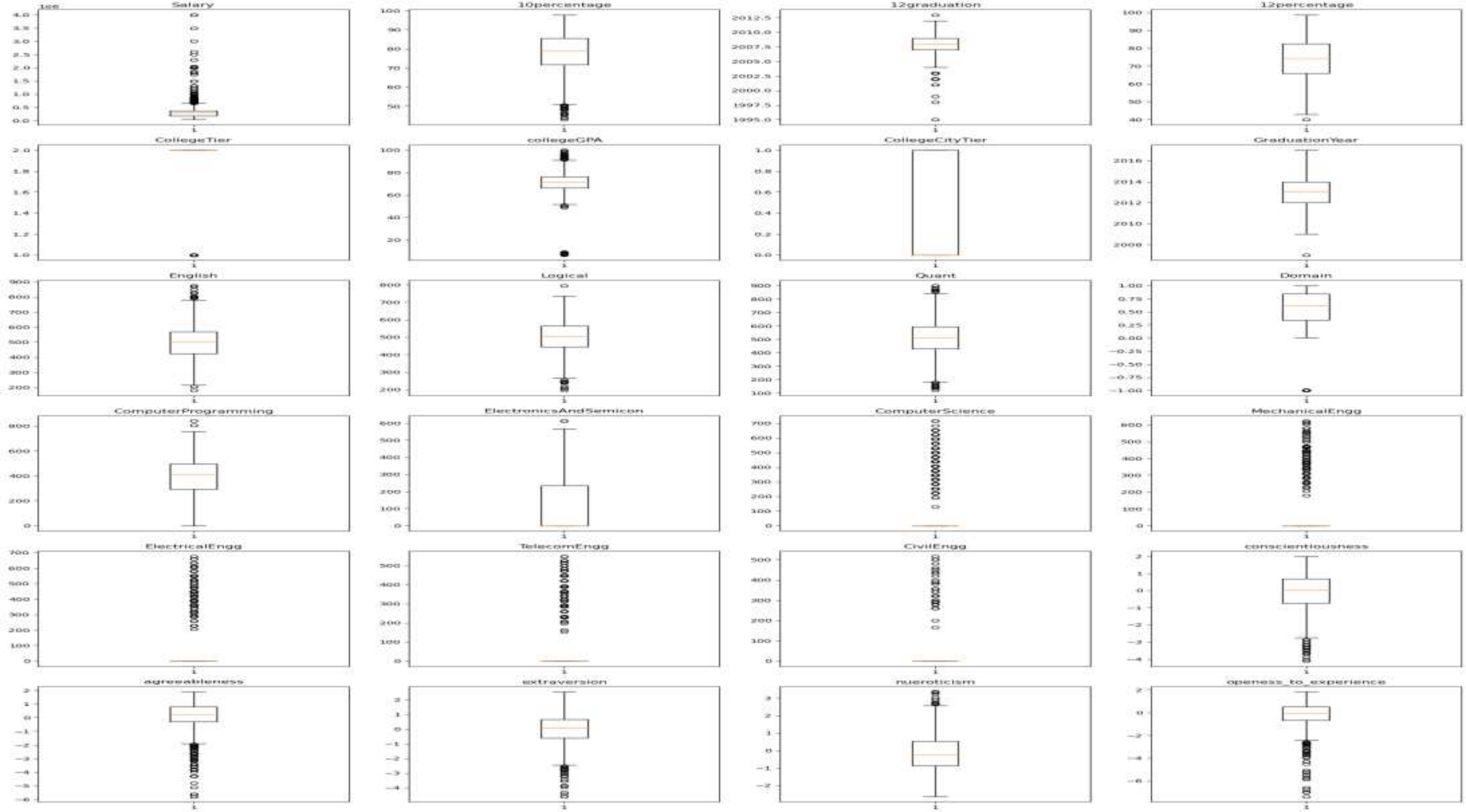


INNOMATICS RESEARCH LABS

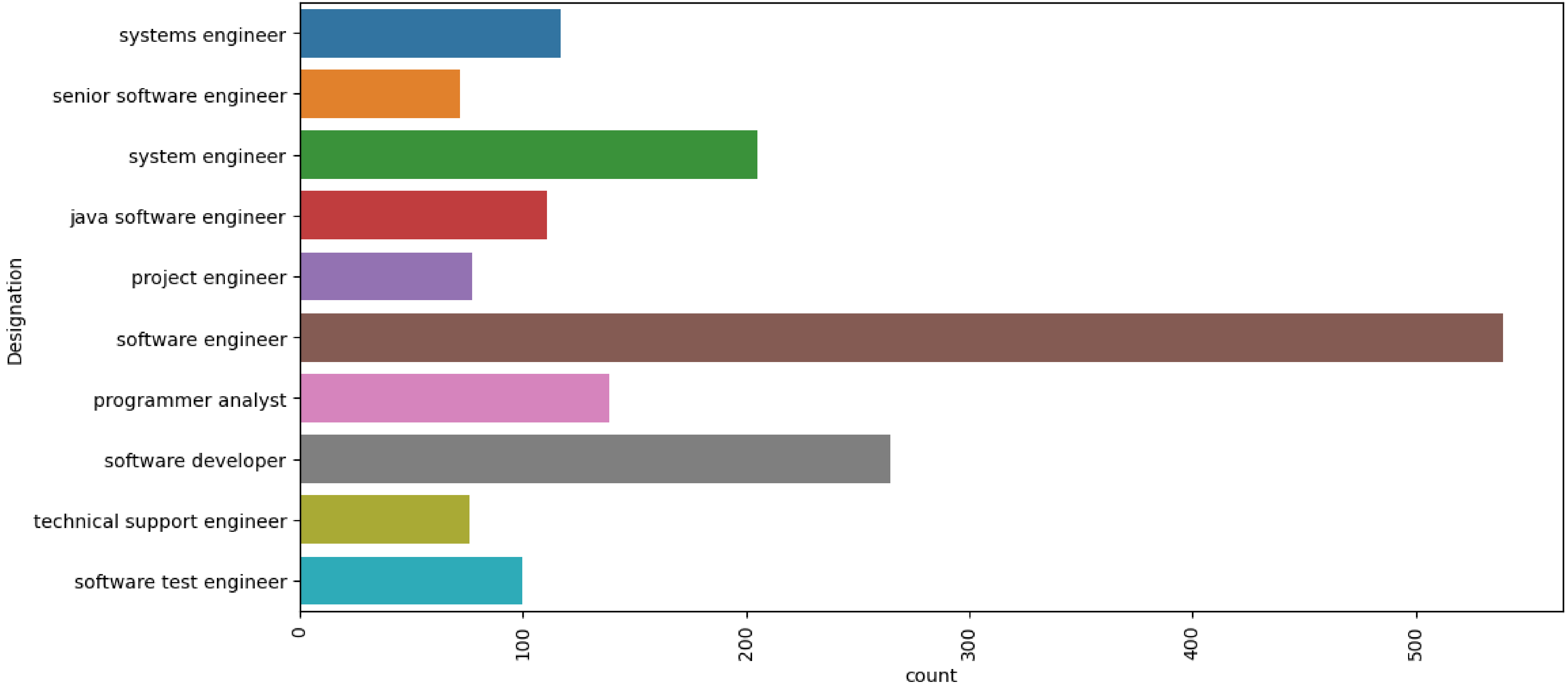
# Hist Plot(For Numerical Columns)



# Box-plot(For Numerical Columns)

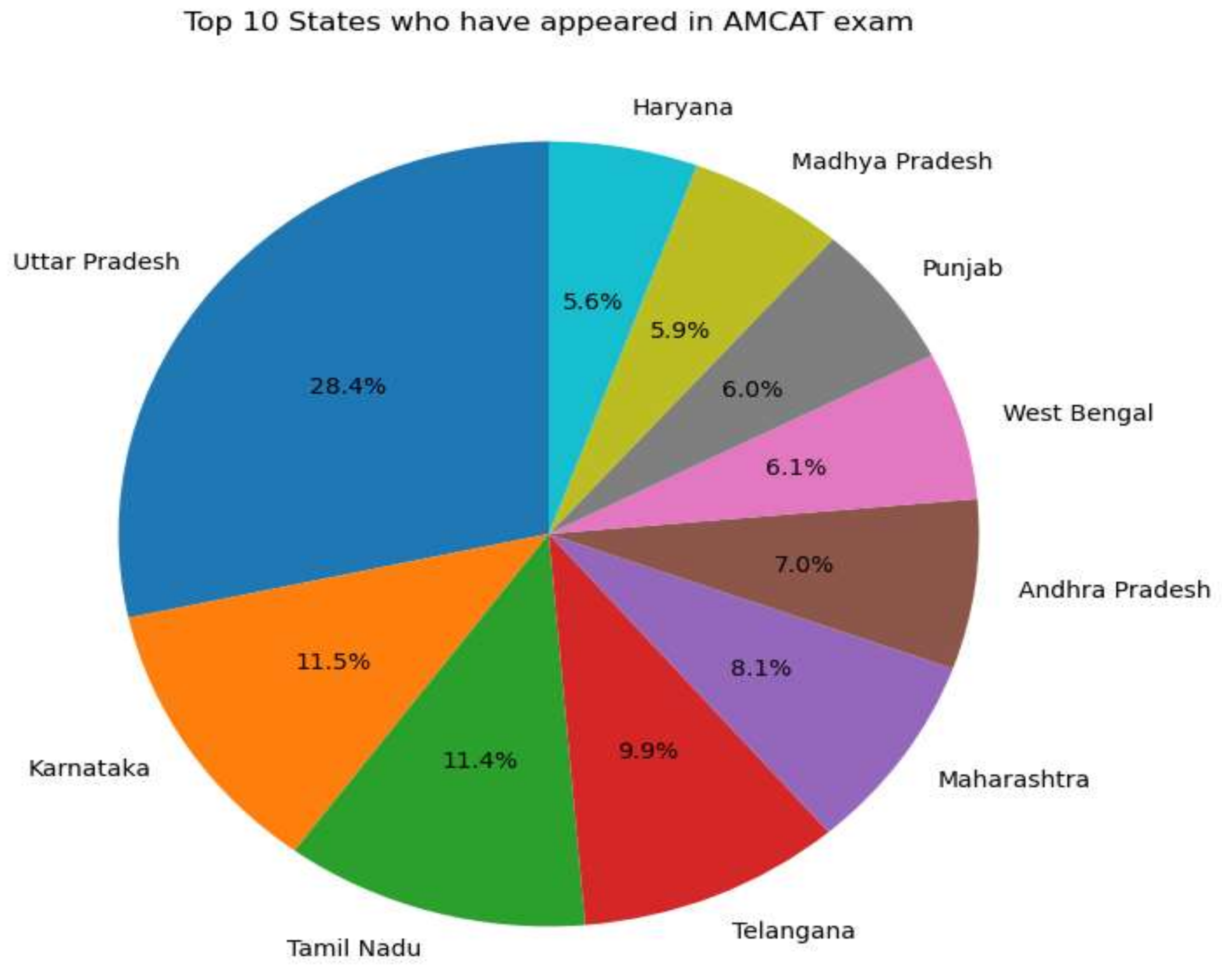


- Similarly we can do the plots for categorical columns also...



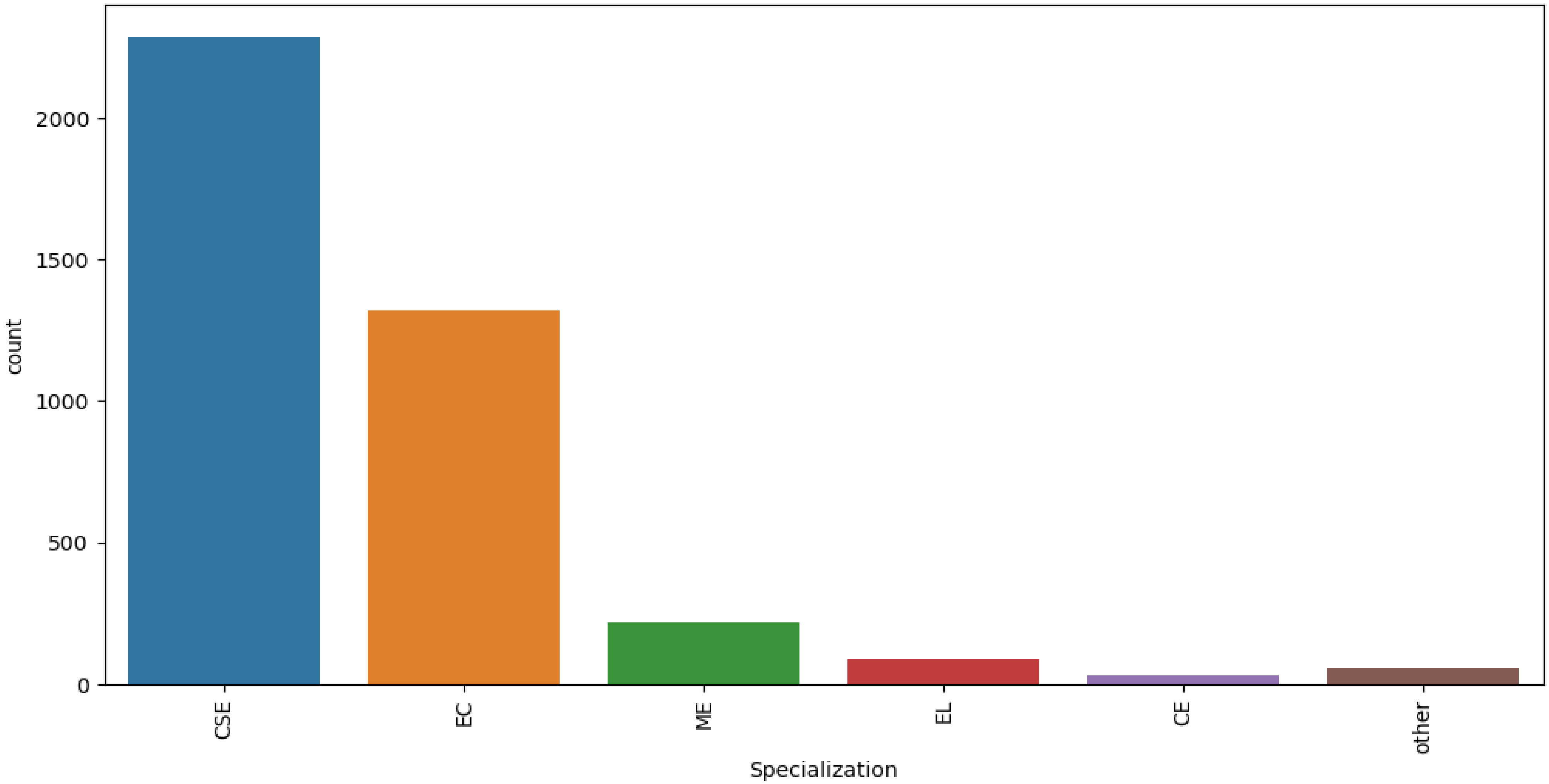
- Observations: Most of the applicants for AMCAT 2015 are working as Software Engineer.

INNOMATICS RESEARCH LABS



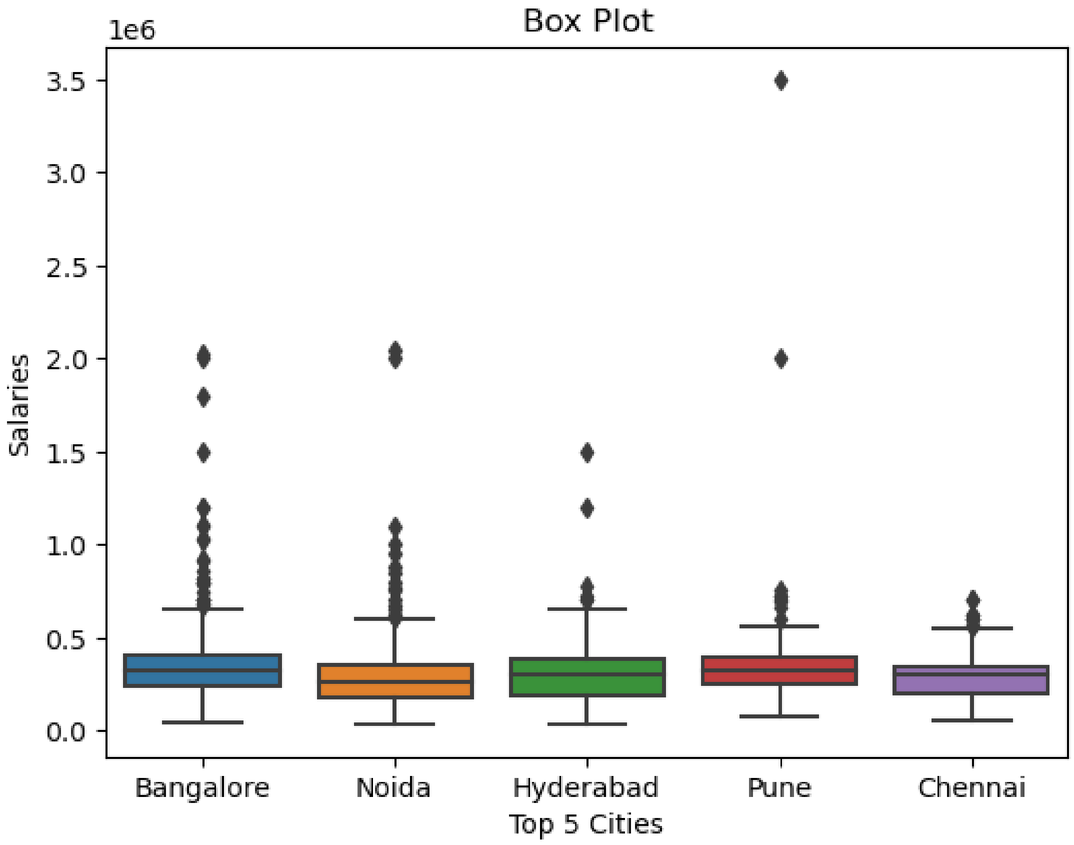
- Observations: Most of the people who have appeared for AMCAT-2015 are from Uttar Pradesh

INNOMATICS RESEARCH LABS

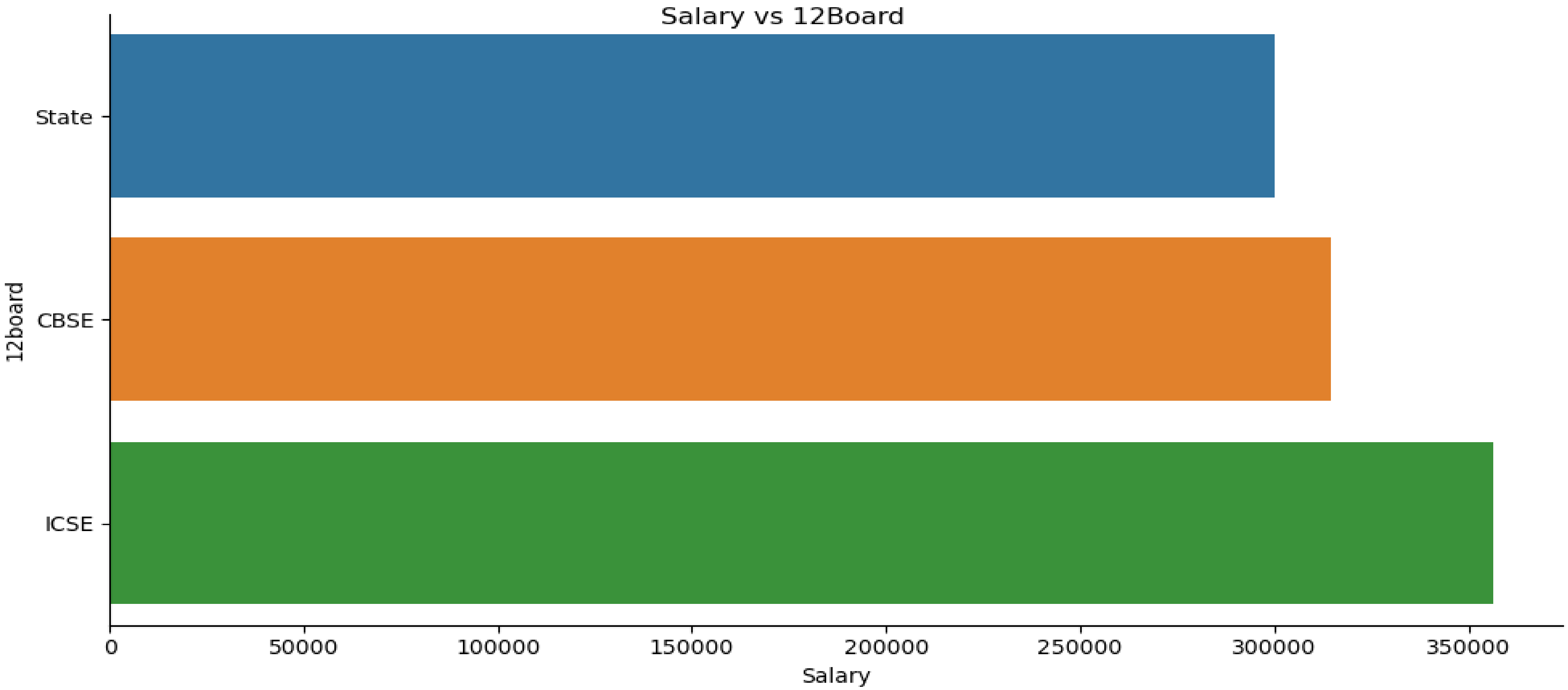


- Observation:Most of the students are from Computer Science background.

# *Bivariate Analysis  Steps*



- Observations: Top 5 Cities(based on workforce) and their salaries



- Observations: From this plot we can say that students who have passed out of 12th board of ICSE

# THANK YOU

# AMCAT Data Analysis

AMCAT (Aspiring Minds Computer Adaptive Test) is an employability assessment test used by companies to evaluate the job-readiness of candidates. The test assesses various skills such as aptitude, technical knowledge, and communication skills.

In [235]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import warnings
warnings.filterwarnings('ignore')
```

In [236]:
```
df=pd.read_csv(r"C:\Users\ASUS\Downloads\data.xlsx - Sheet1.csv")
```

In [237]:
```
df.drop('Unnamed: 0',axis=1,inplace=True)
```

In [238]:
```
df.shape
```

Out[238]:
```
(3998, 38)
```

In [239]:
```
df.head()
```

Out[239]:

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

In [240]:
```
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
```

In [241]:
```
df.describe(include='object')
```

Out[241]:

| | DOJ | DOL | Designation | JobCity | Gender | DOB | 10board | 12board | Degree | Speci |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 3998 | 3998 | 3998 | 3998 | 3998 | 3998 | 3998 | 3998 | 3998 | |
| **unique** | 81 | 67 | 419 | 339 | 2 | 1872 | 275 | 340 | 4 | |
| **top** | 7/1/14 0:00 | present | software engineer | Bangalore | m | 1/1/91 0:00 | cbse | cbse | B.Tech/B.E. | electr comm en |
| **freq** | 199 | 1875 | 539 | 627 | 3041 | 11 | 1395 | 1400 | 3700 | |

In [242]:
```
df.describe(include='number')
```

Out[242]:
```
                 ID        Salary  10percentage  12graduation  12percentage     CollegeID  Colleg
count  3.998000e+03  3.998000e+03   3998.000000   3998.000000   3998.000000   3998.000000  3998.0
 mean  6.637945e+05  3.076998e+05     77.925443   2008.087544     74.466366   5156.851426    1.92
  std  3.632182e+05  2.127375e+05      9.850162      1.653599     10.999933   4802.261482    0.26
  min  1.124400e+04  3.500000e+04     43.000000   1995.000000     40.000000      2.000000    1.0
  25%  3.342842e+05  1.800000e+05     71.680000   2007.000000     66.000000    494.000000    2.0
  50%  6.396000e+05  3.000000e+05     79.150000   2008.000000     74.400000   3879.000000    2.0
  75%  9.904800e+05  3.700000e+05     85.670000   2009.000000     82.600000   8818.000000    2.0
  max  1.298275e+06  4.000000e+06     97.760000   2013.000000     98.700000  18409.000000    2.0
```

In [243]:
```
col=list(df.drop(columns=['ID','CollegeID','CollegeCityID'],axis=1).select_dtypes
```

In [244]:
```
col
```

Out[244]:
```
['Salary',
 '10percentage',
 '12graduation',
 '12percentage',
 'CollegeTier',
 'collegeGPA',
 'CollegeCityTier',
 'GraduationYear',
 'English',
 'Logical',
 'Quant',
 'Domain',
 'ComputerProgramming',
 'ElectronicsAndSemicon',
 'ComputerScience',
 'MechanicalEngg',
 'ElectricalEngg',
 'TelecomEngg',
 'CivilEngg',
 'conscientiousness',
 'agreeableness',
 'extraversion',
 'nueroticism',
 'openess_to_experience']
```

In [245]:
```
pd.set_option('display.max_columns',828)
```

In [246]:

```
df.head()
```

Out[246]:

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

In [247]:

```
out_dict={}
for i in col:
    Q1 = df[i].quantile(0.05)
    Q3 = df[i].quantile(0.95)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[i] < lower_bound) | (df[i] > upper_bound)]
    out_dict[i]=outliers
```

In [248]:

```
len_out={}
for i in col:
    Q1 = df[i].quantile(0.05)
    Q3 = df[i].quantile(0.95)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[i] < lower_bound) | (df[i] > upper_bound)]
    len_out[i]=len(outliers)
```

In [249]:
```
len_out
```

Out[249]:
```
{'Salary': 23,
 '10percentage': 0,
 '12graduation': 1,
 '12percentage': 0,
 'CollegeTier': 0,
 'collegeGPA': 12,
 'CollegeCityTier': 0,
 'GraduationYear': 1,
 'English': 0,
 'Logical': 0,
 'Quant': 0,
 'Domain': 0,
 'ComputerProgramming': 0,
 'ElectronicsAndSemicon': 0,
 'ComputerScience': 0,
 'MechanicalEngg': 0,
 'ElectricalEngg': 161,
 'TelecomEngg': 0,
 'CivilEngg': 42,
 'conscientiousness': 0,
 'agreeableness': 0,
 'extraversion': 0,
 'nueroticism': 0,
 'openess_to_experience': 7}
```

In [250]:
```
out_dict['collegeGPA']
```

Out[250]:

| | ID | Salary | DOJ | DOL | Designation | JobCity | Gender | DOB | 10percentage |
|---|---|---|---|---|---|---|---|---|---|
| **7** | 912934 | 400000.0 | 7/1/14 0:00 | 7/1/15 0:00 | mechanical engineer | Bangalore | m | 5/27/92 0:00 | 92.00 |
| **138** | 964319 | 195000.0 | 10/1/14 0:00 | 1/1/15 0:00 | business development managerde | coimbatore | m | 5/4/91 0:00 | 79.60 |
| **788** | 249853 | 180000.0 | 5/1/12 0:00 | 6/1/13 0:00 | electrical project engineer | Jowai | m | 1/12/89 0:00 | 66.50 |
| **1419** | 1262900 | 180000.0 | 10/1/14 0:00 | 4/1/15 0:00 | java software engineer | Chennai | m | 6/14/93 0:00 | 58.90 |
| **1439** | 299447 | 360000.0 | 8/1/11 0:00 | present | assistant professor | AM | m | 12/11/88 0:00 | 73.06 |
| **1767** | 813008 | 180000.0 | 6/1/14 0:00 | 8/1/14 0:00 | it technician | Bhopal | m | 9/21/92 0:00 | 69.00 |
| **2151** | 262814 | 145000.0 | 2/1/12 0:00 | 4/1/13 0:00 | web developer | New Delhi | m | 6/18/88 0:00 | 61.30 |
| **2229** | 868740 | 240000.0 | 1/1/15 0:00 | 4/1/15 0:00 | product development engineer | Chennai | m | 5/1/92 0:00 | 94.40 |
| **2293** | 407736 | 490000.0 | 10/1/12 0:00 | 12/1/14 0:00 | software engineer | -1 | f | 3/18/90 0:00 | 89.60 |
| **2662** | 240465 | 470000.0 | 7/1/11 0:00 | 3/1/15 0:00 | systems engineer | Kolkata | m | 2/15/90 0:00 | 77.38 |
| **2691** | 385442 | 820000.0 | 7/1/14 0:00 | 3/1/15 0:00 | software engineer | New Delhi | m | 10/28/90 0:00 | 81.20 |
| **3308** | 287976 | 250000.0 | 8/1/11 0:00 | 11/1/12 0:00 | engineer | Aurangabad | m | 6/7/85 0:00 | 63.20 |

In [251]:
```
repl=['ComputerProgramming','ElectronicsAndSemicon','ComputerScience','Mechanical
```

In [252]:
```
for i in repl:
    df[i]=df[i].replace(-1,np.nan)
```

In [253]:
```
out_dict['GraduationYear']['GraduationYear'].iloc[0]
```

Out[253]:
```
0
```

In [254]:
```
df=df[df['GraduationYear']>0]
```

In [255]:
```
len(df[df['collegeGPA']<=10])
```

Out[255]:
```
12
```

In [256]:
```
len(df[df['collegeGPA']>10])
```

Out[256]:
```
3985
```

In [257]:
```
#df=df[df['collegeGPA']>10]
```

In [258]:
```
out_dict['12graduation']
```

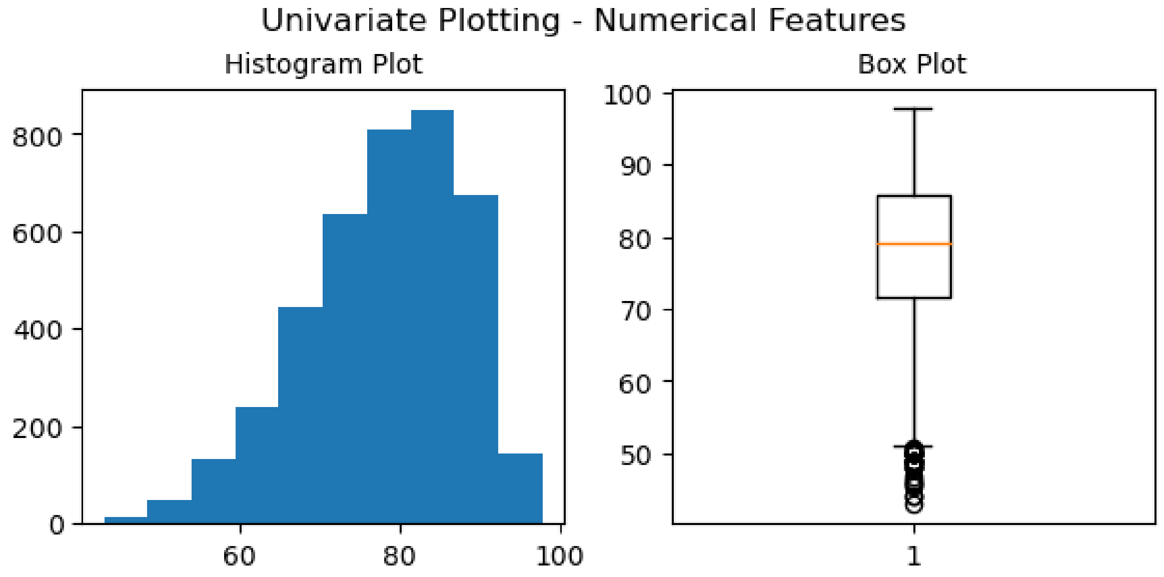
Out[258]:

| | ID | Salary | DOJ | DOL | Designation | JobCity | Gender | DOB | 10percentage | 10boar |
|---|---|---|---|---|---|---|---|---|---|---|
| **59** | 536053 | 120000.0 | 9/1/09 0:00 | 4/1/13 0:00 | software engineer | Bangalore | m | 10/30/77 0:00 | 72.0 | cbs |

In [259]:
```
len(col)
```

Out[259]:
```
24
```

In [260]:
```
fig, axs = plt.subplots(1, 2, figsize=(6, 3), layout="constrained")
fig.suptitle("Univariate Plotting - Numerical Features")

axs[0].hist(df["10percentage"])
axs[0].set_title("Histogram Plot", fontsize="medium")

axs[1].boxplot(df["10percentage"])
axs[1].set_title("Box Plot", fontsize="medium")

plt.show()
```



In [ ]:
```

```

In [267]:
```
df['JobCity'].replace({'Vizag':'Visakhapatnam','VIZAG':'Visakhapatnam','vizag':'V
```

In [268]:
```
#!pip install fuzzywuzzy
```

In [269]:

```
from fuzzywuzzy import process

def correct_spelling_errors(target_word,choices=['Bangalore', 'Indore', 'Chennai'
                                                 'Hyderabad', 'Noida', 'Kolkata',
                                                 'Bhubaneswar', 'Mumbai', 'New De
                                                 'Mangalore', 'Rewari', 'Ghaziaba
                                                 'Jaipur', 'Thane', 'Maharajganj'
                                                 'Coimbatore', 'Dhanbad', 'Luckno
                                                 'Nagpur', 'Bhagalpur', 'New Delh
                                                 'Bankura', 'Kanpur', 'Vijayawada
                                                 'Bhopal', 'Faridabad', 'Jodhpur'
                                                 'Haridwar', 'Raigarh', 'Visakhap
                                                 'Belgaum', 'Dehradun', 'Rudrapur
                                                 'Hissar', 'Ranchi', 'Madurai', '
                                                 'Jagdalpur', 'Angul', 'Baroda',
                                                 'Neemrana', 'Tirupati', 'Calicut
                                                 'Gagret', 'Indirapuram, Ghaziaba
                                                 'Hospet', 'Miryalaguda', 'Dharuh
                                                 'Agra', 'Trichy', 'Kudankulam ,
                                                 'Sadulpur', 'Bikaner', 'Vadodara
                                                 'Tirunelveli', 'Ernakulam', 'Bi
                                                 'Patna','Salem','Technopark, Thi
                                                 'Shimla', 'Jammu', 'Shahdol','Mu
                                                 'Ratnagiri', 'Jhajjar', 'Gulbarg
                                                 'Odisha', 'Kharagpur', 'Navi Mum
                                                 'Karnal','London', 'Kota', 'Badd
                                                 'Rayagada, Odisha', 'Kakinada',
                                                 'Sahibabad', 'Howrah', 'Trichur'
                                                 'Delhi/NCR', 'Jalandhar', 'Manes
                                                 'Phagwara', 'Baripada', 'Yamunan
                                                 'Latur', 'Mainpuri', 'Rae Bareli
                                                 'Karad', 'Rajpura', 'Haryana'], 
    match, score = process.extractOne(target_word, choices)
    if score >= threshold:
        return match
    else:
        return target_word
```

In [270]:

```
df['JobCity']=df['JobCity'].apply(correct_spelling_errors)
```

In [271]:
```
df['JobCity'].unique()
```

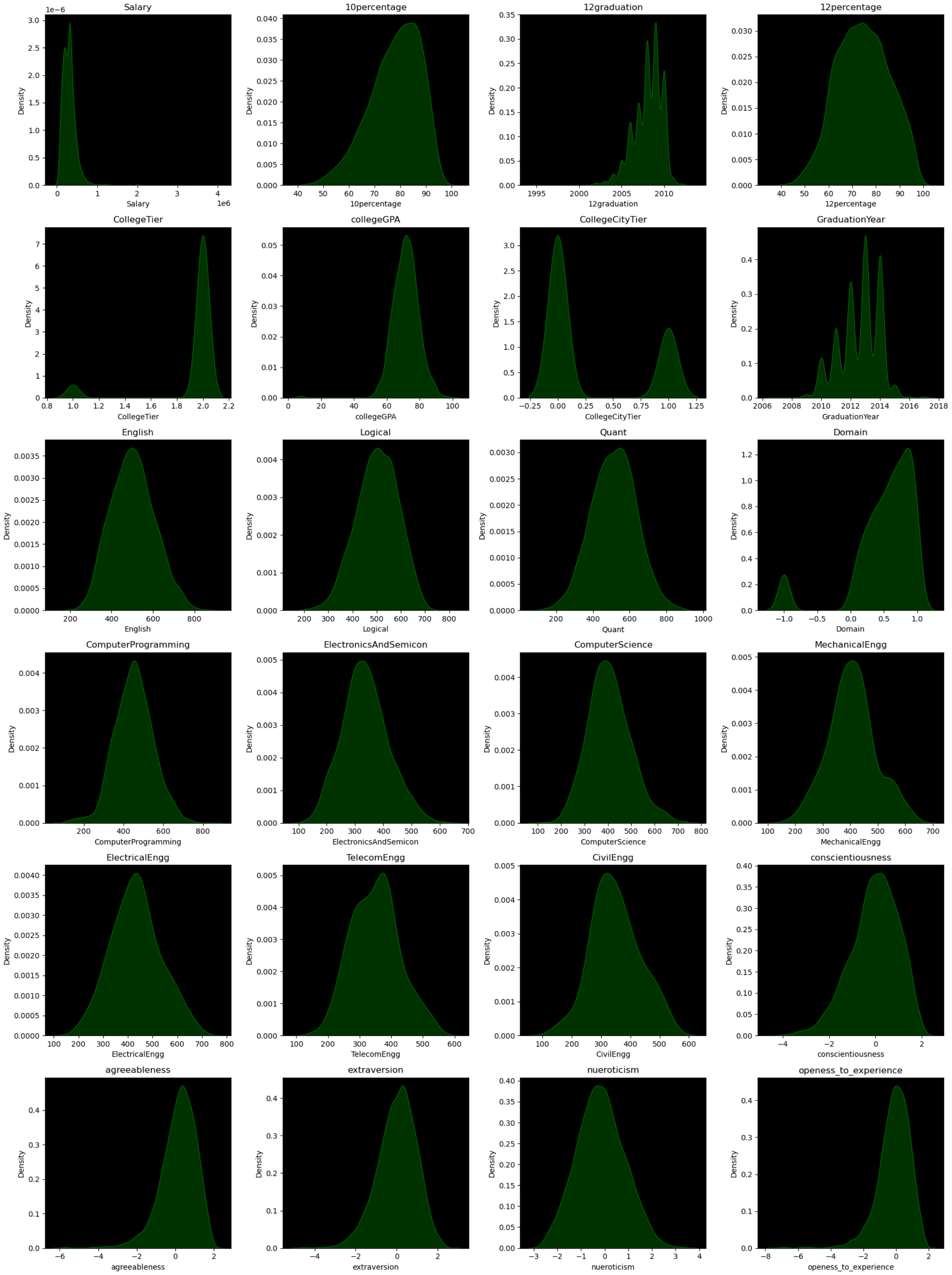
Out[271]:
```
array(['Bangalore', 'Indore', 'Chennai', 'Gurgaon', 'Manesar',
       'Hyderabad', 'Noida', 'Kolkata', 'Pune', '-1', 'Mohali', 'Jhansi',
       'New Delhi', 'Bhubaneswar', 'Mumbai', 'Mangalore', 'Rewari',
       'Ghaziabad', 'Bhiwadi', 'Mysore', 'Rajkot', 'Jaipur', 'Thane',
       'Maharajganj', 'Thiruvananthapuram', 'Panchkula', 'Coimbatore',
       'Dhanbad', 'Lucknow', 'Gandhinagar', 'Una', 'Daman and Diu',
       'Visakhapatnam', 'Nagpur', 'Bhagalpur', 'New Delhi/Jaisalmer',
       'Ahmedabad', 'Kochi/Cochin', 'Bankura', 'Kanpur', 'Vijayawada',
       'Beawar', 'Alwar', 'Siliguri', 'Raipur', 'Bhopal', 'Faridabad',
       'Jodhpur', 'Udaipur', 'Muzaffarpur', 'Bulandshahar', 'Haridwar',
       'Raigarh', 'Jabalpur', 'Unnao', 'Aurangabad', 'Belgaum',
       'Dehradun', 'Rudrapur', 'Jamshedpur', 'Dharamshala', 'Hissar',
       'Ranchi', 'Madurai', 'Chandigarh', 'Australia', 'Cheyyar',
       'Sonipat', 'Nagari', 'Jagdalpur', 'Angul', 'Baroda', ' Ariyalur',
       'Jowai', 'Kochi/Cochin, Chennai and Coimbatore', 'Neemrana',
       'Tirupati', 'Calicut', 'Dubai', 'bengaluru', 'Ahmednagar',
       'Nashik', 'Bellary', 'Ludhiana', 'Muzaffarnagar', 'Gagret',
       'Indirapuram, Ghaziabad', 'Gwalior', 'Chennai & Mumbai',
       'Rajasthan', 'Bareli', 'Hospet', 'Miryalaguda', 'Dharuhera',
       'Meerut', 'Ganjam', 'Hubli', 'Agra', 'Trichy',
       'Kudankulam , Tarapur', 'Ongole', 'Sambalpur', 'Pondicherry',
       'Bundi', 'N/A', 'Bikaner', 'Vadodara', 'India', 'Asansol',
       'Tirunelvelli', 'Ernakulam', 'Bilaspur', 'Chandrapur', 'Nanded',
       'Dharmapuri', 'Vandavasi', 'Rohtak', 'trivandrum', 'Patna',
       'Salem', 'Technopark, Thiruvananthapuram', 'Bharuch', 'Tornagallu',
       'Jaspur', 'Burdwan', 'Shimla', 'Jammu', 'Shahdol', 'Muvattupuzha',
       'Al Jubail', 'Kalmar, Sweden', 'Secunderabad', 'Ratnagiri',
       'Jhajjar', 'Gulbarga', 'Nalagarh', 'Jeddah', 'Jamnagar', 'Gonda',
       'Odisha', 'Kharagpur', 'Navi Mumbai , Hyderabad', 'Joshimath',
       'Bathinda', 'Johannesburg', 'Kala Amb', 'Karnal', 'London', 'Kota',
       'Baddi', 'Mettur', 'Durgapur', 'Surat', 'Kurnool', 'Kolhapur',
       'Bhilai', 'Bahadurgarh', 'Rayagada, Odisha', 'Kakinada',
       'Varanasi', 'Nellore', 'Sahibabad', 'Howrah', 'Trichur', 'Ambala',
       'Khopoli', 'Kerala', 'Roorkee', 'Allahabad', 'Delhi/NCR',
       'Jalandhar', 'Vapi', 'Pilani', 'Ras Al Khaimah', 'Bihar',
       'Singaruli', 'Phagwara', 'Baripada', 'Yamunanagar', 'Shahibabad',
       'Sampla', 'Guwahati', 'Rourkela', 'Vellore', 'Dausa', 'Latur',
       'Mainpuri', 'Dammam', 'Haldia', 'Rae Bareli', 'Patiala',
       'Gorakhpur', 'Karad', 'Rajpura', 'Haryana'], dtype=object)
```

# Univariate Analysis

## For Numerical Columns

### KDE

In [272]:
```
fig, axes = plt.subplots(6, 4, figsize=(18, 24))
axes = axes.flatten()
for i, column in enumerate(col):
    sns.kdeplot(data=df[column], ax=axes[i], label=column,fill=True, color='darkg
    axes[i].set_title(column)  # Set subplot title
    axes[i].set_facecolor('black')
for ax in axes[len(col):]:
    ax.axis('off')
plt.tight_layout()
plt.show()
```

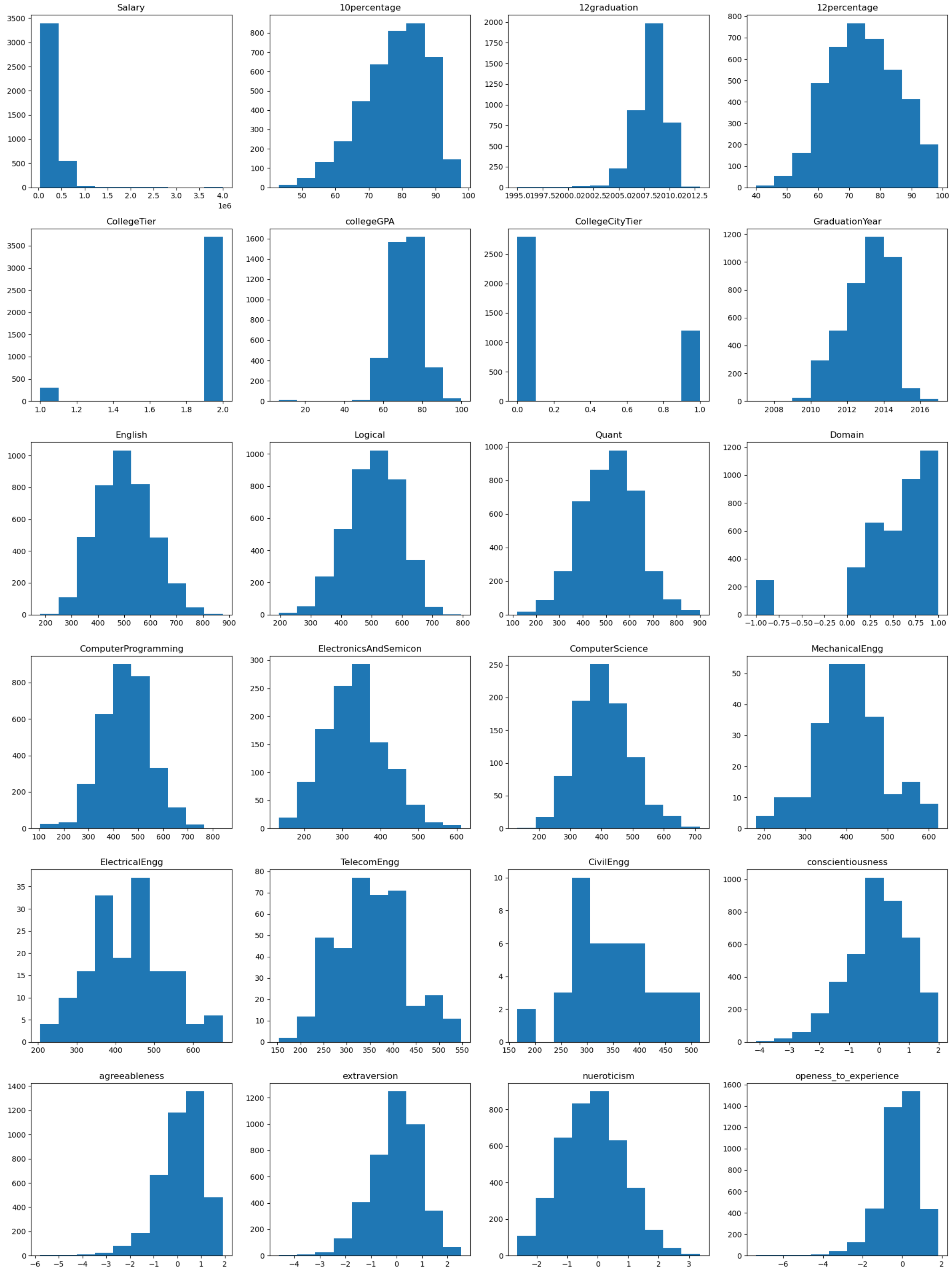


- These are the KDE(Kernel Density Estimation) plots for numerical columns
- We can see the trends for different columns.

### Histograms

In [273]:

```
fig, axes = plt.subplots(6, 4, figsize=(18, 24))
axes = axes.flatten()
for i, column in enumerate(col):
    axes[i].hist(df[column], bins=10)
    axes[i].set_title(column)
for ax in axes[len(col):]:
    ax.axis('off')
plt.tight_layout()
plt.show()
```

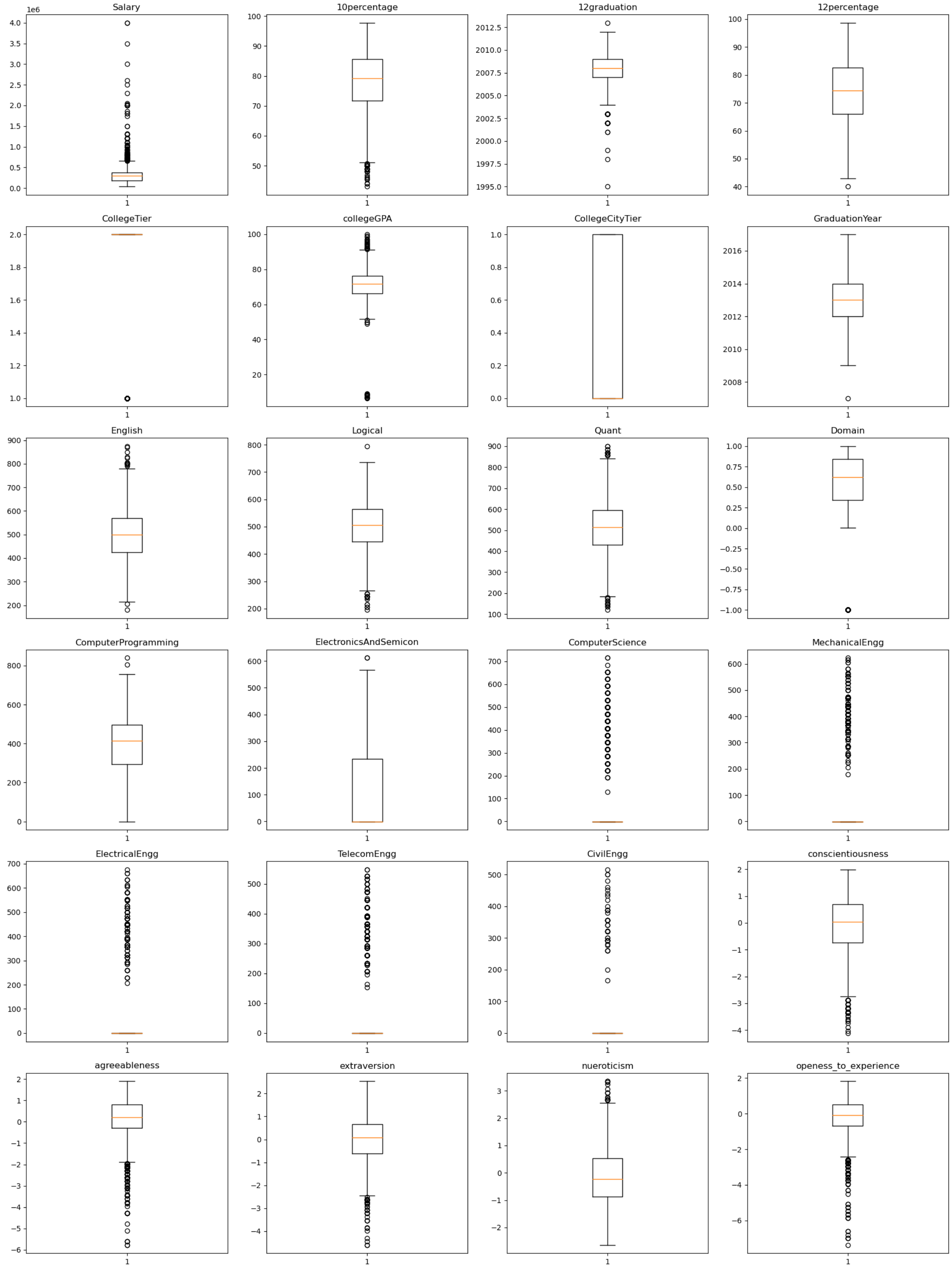


- These are the Histogram plots for Numerical columns

### Box-Plots

In [274]:

```python
fig, axes = plt.subplots(6, 4, figsize=(18, 24))
df_filled = df.fillna(-1)
axes = axes.flatten()
for i, column in enumerate(col):
    axes[i].boxplot(df_filled[column])
    axes[i].set_title(column)
for ax in axes[len(col):]:
    ax.axis('off')
plt.tight_layout()
plt.show()
```



- These are the box plots reperesenting the data and outliers(if available even after cleaning).

### For Categorical columns

In [275]:
```
pd.Timestamp.now()
```

Out[275]:
```
Timestamp('2024-02-23 09:58:31.864478')
```

In [276]:
```
import warnings
warnings.filterwarnings("ignore")

from datetime import datetime
datetime.now()
```

Out[276]:
```
datetime.datetime(2024, 2, 23, 9, 58, 31, 887822)
```

In [277]:
```
df['DOJ']=pd.to_datetime(df['DOJ'])
```

In [278]:
```
df['DOB']=pd.to_datetime(df['DOB'])
```

In [279]:
```
df['DOL']=df['DOL'].replace('present',pd.Timestamp.now())
```

In [280]:
```
df['DOL']=pd.to_datetime(df['DOL'])
```

In [281]:
```
df['JobCity'].value_counts(dropna=False)
```

Out[281]:
```
Bangalore              685
-1                     461
Noida                  420
Hyderabad              370
Pune                   328
                      ... 
Nanded                   1
New Delhi/Jaisalmer      1
Bankura                  1
Ernakulam                1
Haryana                  1
Name: JobCity, Length: 195, dtype: int64
```

In [282]:
```
df['JobCity']=df['JobCity'].replace('-1','N/A')#.value_counts(dropna=False)
```

In [283]:
```
df['JobCity'].value_counts(dropna=False)
```

Out[283]:
```
Bangalore              685
N/A                    462
Noida                  420
Hyderabad              370
Pune                   328
                      ... 
Dharmapuri               1
Nanded                   1
New Delhi/Jaisalmer      1
Bankura                  1
Haryana                  1
Name: JobCity, Length: 194, dtype: int64
```

In [284]:
```
df['10board'].unique()
```

```
       'karnataka secondary education examination board', 'delhi board',
       'mirza ahmed ali baig', 'jseb', 'bse, odisha', 'bihar board',
       'maharashtra state(latur board)', 'rajasthan board', 'mpboard',
       'upbhsie', 'secondary board of rajasthan',
       'tamilnadu matriculation board', 'jharkhand secondary board',
       'board of secondary education,andhara pradesh', 'up baord',
       'state', 'board of intermediate education',
       'state board of secondary education,andhra pradesh',
       'up board , allahabad',
       'stjosephs girls higher sec school,dindigul', 'maharashtra board',
       'education board of kerala', 'board of ssc',
       'maharashtra state board pune',
       'board of school education harayana',
       'secondary school cerfificate', 'maharashtra sate board', 'ksseb',
       'bihar examination board, patna', 'latur',
       'board of secondary education, rajasthan', 'state borad hp',
       'cluny', 'bsepatna', 'up borad', 'ssc board of andrapradesh',
       'matric', 'bse,orissa', 'ssc-andhra pradesh', 'mp',
       'karnataka education board', 'mhsbse',
       'karnataka sslc board bangalore', 'karnataka', 'u p',
```

In [285]:
```
df['10board']=df['10board'].replace('0','N/A')
```

In [286]:
```
df['12board'].unique()
```

Out[286]:
```
array(['board of intermediate education,ap', 'cbse', 'state board',
       'mp board', 'isc', 'icse', 'karnataka pre university board', 'up',
       'p u board, karnataka', 'dept of pre-university education', 'bie',
       'kerala state hse board', 'up board', '0', 'bseb', 'chse', 'puc',
       ' upboard',
       'state  board of intermediate education, andhra pradesh',
       'karnataka state board',
       'west bengal state council of technical education', 'wbchse',
       'maharashtra state board', 'ssc', 'isc board',
       'sda matric higher secondary school', 'uttar pradesh board', 'ibe',
       'chsc', 'board of intermediate', 'isce', 'upboard', 'sbtet',
       'hisher seconadry examination(state board)', 'pre university',
       'borad of intermediate', 'j & k board',
       'intermediate board of andhra pardesh', 'rbse',
       'central board of secondary education', 'jkbose', 'hbse',
       'board of intermediate education', 'state', 'ms board', 'pue',
       'intermediate state board', 'stateboard', 'hsc',
       'electonincs and communication(dote)', 'karnataka pu board',
       'government polytechnic mumbai , mumbai board', 'pu board',
       'board of intermediate education', 'apbie', 'andhra board',
```

In [287]:
```
df['12board']=df['12board'].replace('0','N/A')
```

In [288]:
```
df['Designation'].value_counts(dropna=False)
```

Out[288]:
```
software engineer                      539
software developer                     265
system engineer                        205
programmer analyst                     139
systems engineer                       117
                                      ... 
cad drafter                              1
noc engineer                             1
human resources intern                   1
senior quality assurance engineer        1
jr. software developer                   1
Name: Designation, Length: 419, dtype: int64
```

In [289]:
```
board10=list(df['10board'].unique())
```

In [290]:
```
board12=list(df['12board'].unique())
```

In [291]:
```
state_10=[]
cbse_10=[]
icse_10=[]
for i in board10:
    if i in ('cbse','cbse[gulf_zone]','cbse ','cbsc','new delhi','board of second
        cbse_10.append(i)
    elif i in ('icse','icse board','cicse'):
        icse_10.append(i)
    else:
        state_10.append(i)
```

In [292]:
```
for i in state_10:
    df['10board'].replace(i,'State',inplace=True)
for i in cbse_10:
    df['10board'].replace(i,'CBSE',inplace=True)
for i in icse_10:
    df['10board'].replace(i,'ICSE',inplace=True)
```

In [293]:
```
state_12=[]
cbse_12=[]
icse_12=[]
for i in board12:
    if i in ('cbse','cbese ','cbsc','new delhi','cbse board','bice'):
        cbse_12.append(i)
    elif i in ('icse','ise board','cicse','isce','isc'):
        icse_12.append(i)
    else:
        state_12.append(i)
```

In [294]:
```
for i in state_12:
    df['12board'].replace(i,'State',inplace=True)
for i in cbse_12:
    df['12board'].replace(i,'CBSE',inplace=True)
for i in icse_12:
    df['12board'].replace(i,'ICSE',inplace=True)
```

In [295]:
```
df['10board'].value_counts()
```

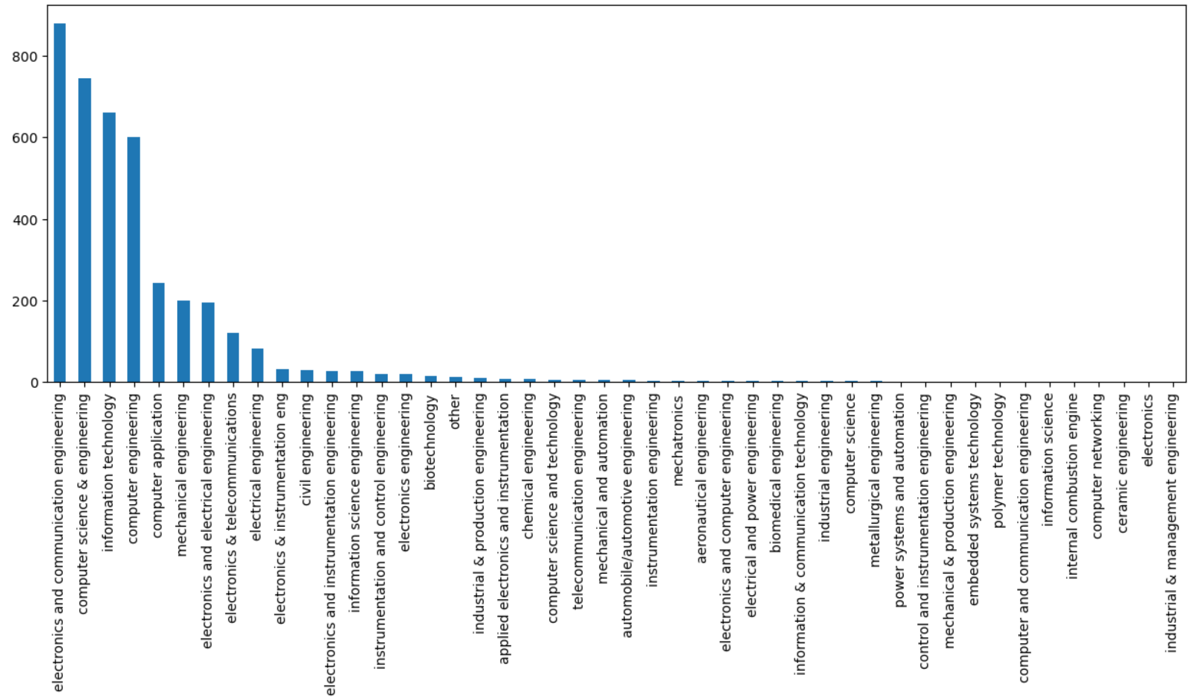
Out[295]:
```
State    2298
CBSE     1416
ICSE      283
Name: 10board, dtype: int64
```

In [296]:
```
df['12board'].value_counts()
```

Out[296]:
```
State    2419
CBSE     1402
ICSE      176
Name: 12board, dtype: int64
```

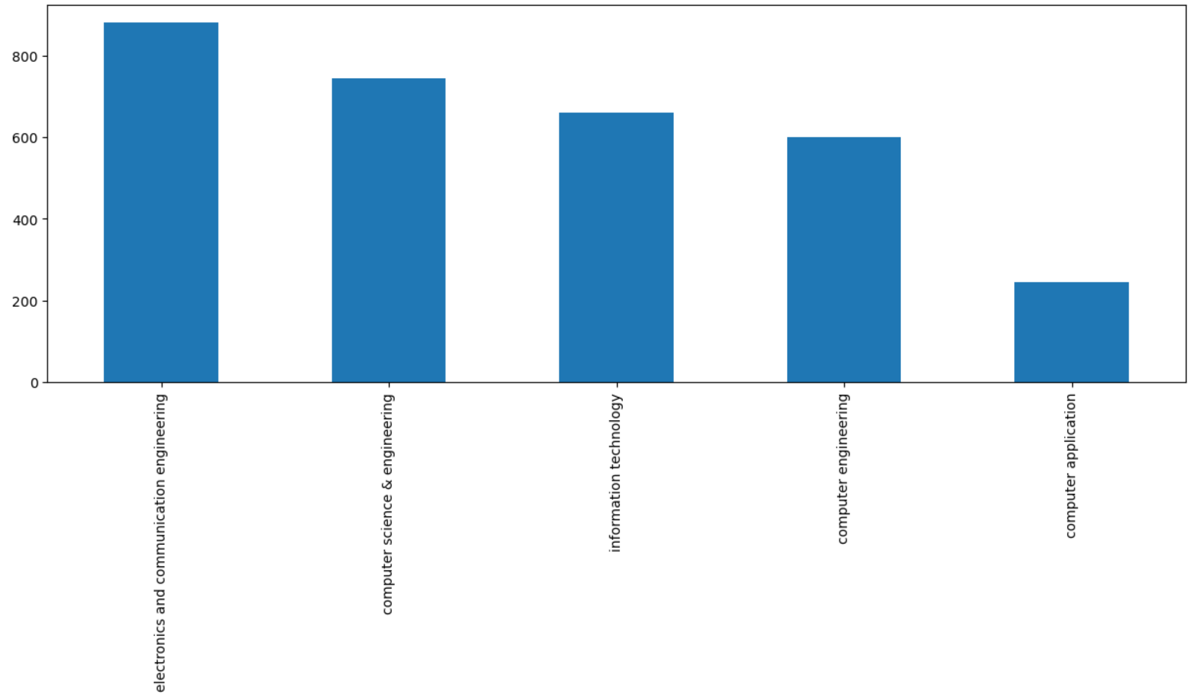
In [298]:
```
specialization_freq = df['Specialization'].value_counts()
specialization_freq.plot(kind='bar', figsize=(15,5))
```

Out[298]:
```
<Axes: >
```



```
In [299]: specialization_freq[0:5].plot(kind='bar', figsize=(15,5))
```

```
Out[299]: <Axes: >
```



In [359]:
```
specialization_map = \
{'electronics and communication engineering' : 'EC',
 'computer science & engineering' : 'CSE',
 'information technology' : 'CSE' ,
 'computer engineering' : 'CSE',
 'computer application' : 'CSE',
 'mechanical engineering' : 'ME',
 'electronics and electrical engineering' : 'EC',
 'electronics & telecommunications' : 'EC',
 'electrical engineering' : 'EL',
 'electronics & instrumentation eng' : 'EC',
 'civil engineering' : 'CE',
 'electronics and instrumentation engineering' : 'EC',
 'information science engineering' : 'CSE',
 'instrumentation and control engineering' : 'EC',
 'electronics engineering' : 'EC',
 'biotechnology' : 'other',
 'other' : 'other',
 'industrial & production engineering' : 'other',
 'chemical engineering' : 'other',
 'applied electronics and instrumentation' : 'EC',
 'computer science and technology' : 'CSE',
 'telecommunication engineering' : 'EC',
 'mechanical and automation' : 'ME',
 'automobile/automotive engineering' : 'ME',
 'instrumentation engineering' : 'EC',
 'mechatronics' : 'ME',
 'electronics and computer engineering' : 'CSE',
 'aeronautical engineering' : 'ME',
 'computer science' : 'CSE',
 'metallurgical engineering' : 'other',
 'biomedical engineering' : 'other',
 'industrial engineering' : 'other',
 'information & communication technology' : 'EC',
 'electrical and power engineering' : 'EL',
 'industrial & management engineering' : 'other',
 'computer networking' : 'CSE',
 'embedded systems technology' : 'EC',
 'power systems and automation' : 'EL',
 'computer and communication engineering' : 'CSE',
 'information science' : 'CSE',
 'internal combustion engine' : 'ME',
 'ceramic engineering' : 'other',
 'mechanical & production engineering' : 'ME',
 'control and instrumentation engineering' : 'EC',
 'polymer technology' : 'other',
 'electronics' : 'EC'}
```

In [360]:
```
df['Specialization'] = df['Specialization'].map(specialization_map)
```

In [300]:
```
df['DOJ']=pd.to_datetime(df['DOJ']).dt.date
df['DOL']=pd.to_datetime(df['DOL']).dt.date
```

In [301]:
```
df['DOJ']=pd.to_datetime(df['DOJ'])
df['DOL']=pd.to_datetime(df['DOL'])
```

In [302]:
```
df['Age']=df['DOJ']-df['DOB']
```

```
In [303]: df['Age']=(df['Age']//365).astype('str')
```

```
In [304]: df['Age']=df['Age'].apply(lambda x: int(re.findall(r'[0-9]+',x)[0]))
```

```
In [305]: df['Age']
```

```
Out[305]: 0       22
          1       23
          2       21
          3       21
          4       23
                  ..
          3993    24
          3994    20
          3995    22
          3996    22
          3997    21
          Name: Age, Length: 3997, dtype: int64
```

```
In [306]: df['Experience']=((df['DOL'].dt.date-df['DOJ'].dt.date)//365).astype('str')
```

```
In [307]: df['Experience']=df['Experience'].apply(lambda x: int(re.findall(r'[0-9]+',x)[0])
```

```
In [308]: df.head()
```

Out[308]:

| | ID | Salary | DOJ | DOL | Designation | JobCity | Gender | DOB | 10percentage | 10board |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 203097 | 420000.0 | 2012-06-01 | 2024-02-23 | senior quality engineer | Bangalore | f | 1990-02-19 | 84.3 | State |
| **1** | 579905 | 500000.0 | 2013-09-01 | 2024-02-23 | assistant manager | Indore | m | 1989-10-04 | 85.4 | CBSE |
| **2** | 810601 | 325000.0 | 2014-06-01 | 2024-02-23 | systems engineer | Chennai | f | 1992-08-03 | 85.0 | CBSE |
| **3** | 267447 | 1100000.0 | 2011-07-01 | 2024-02-23 | senior software engineer | Gurgaon | m | 1989-12-05 | 85.6 | CBSE |
| **4** | 343523 | 200000.0 | 2014-03-01 | 2015-03-01 | get | Manesar | m | 1991-02-27 | 78.0 | CBSE |

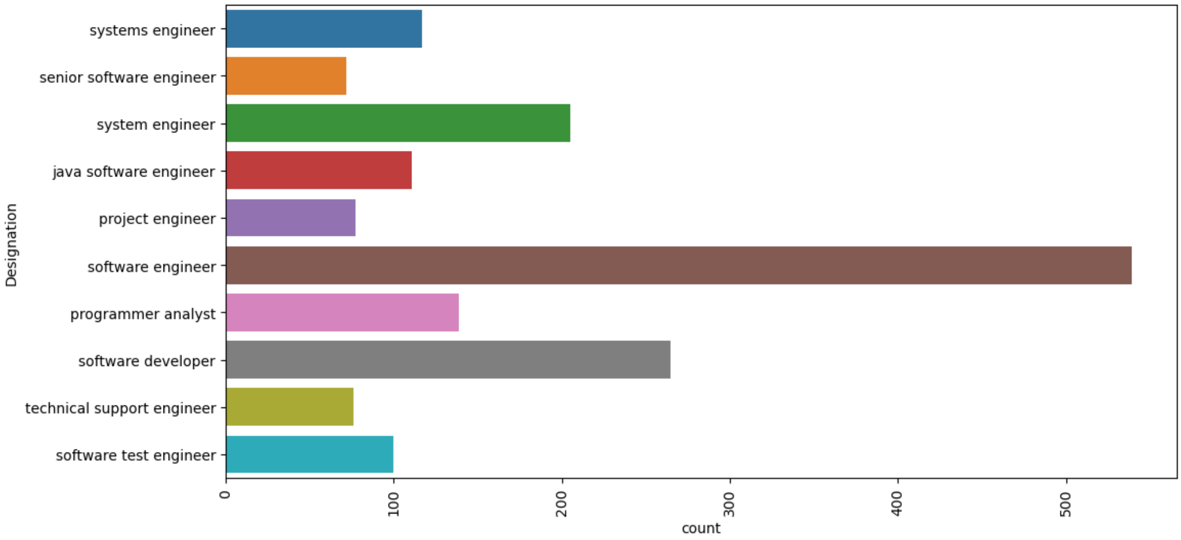
```
In [309]: cat=list(df.select_dtypes(include=['category','object']).columns) ## We are not c
```

```
In [310]: cat
```

```
Out[310]: ['Designation',
           'JobCity',
           'Gender',
           '10board',
           '12board',
           'Degree',
           'Specialization',
           'CollegeState']
```

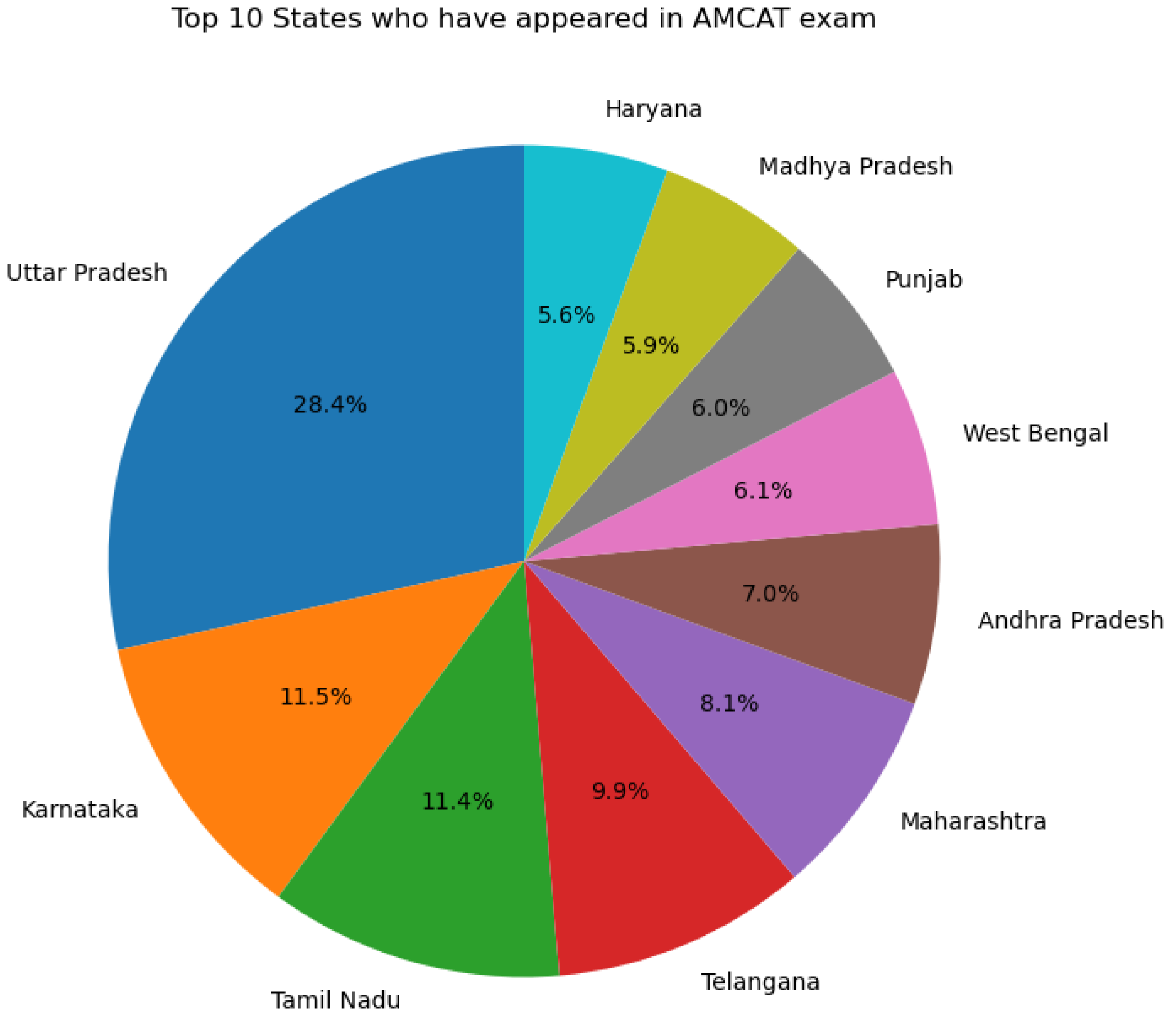
In [ ]:

In [128]:
```
top_values = df['Designation'].value_counts().nlargest(10).index
plt.figure(figsize=(12, 6))
sns.countplot(y='Designation', data=df[df['Designation'].isin(top_values)])
plt.xticks(rotation=90)
plt.show()
```



### Observations: Most of the applicants for AMCAT 2015 are working as Software Engineer.

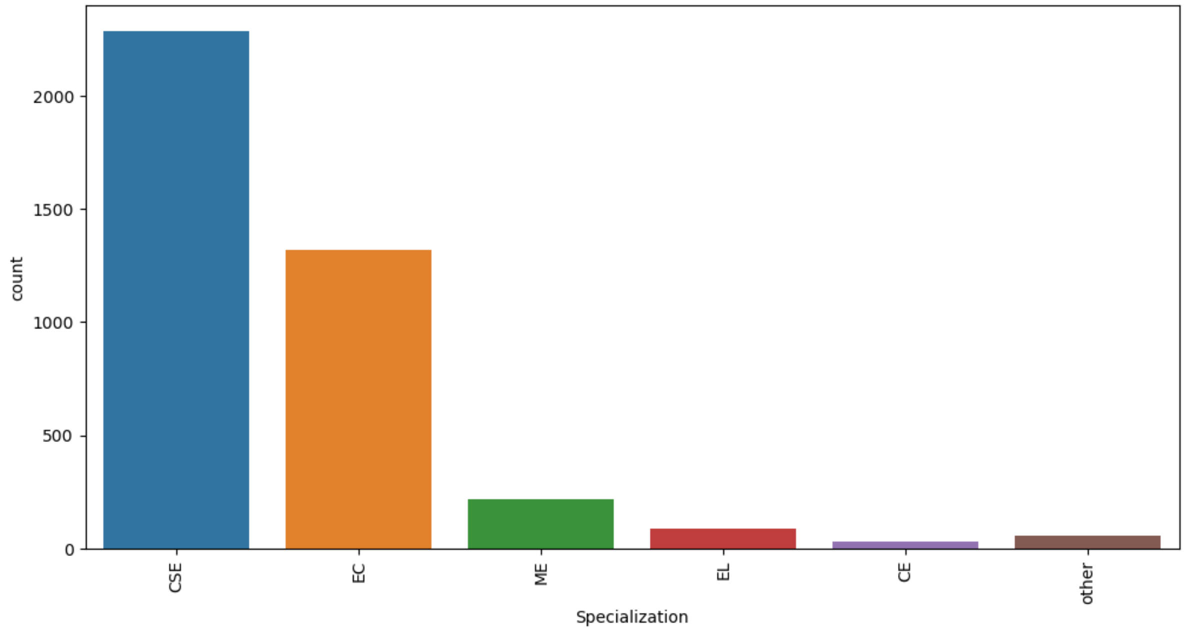
```
In [131]: top_values = df['CollegeState'].value_counts().nlargest(10)
          plt.figure(figsize=(8,8))
          plt.pie(top_values, labels=top_values.index, autopct='%1.1f%%', startangle=90)
          plt.title('Top 10 States who have appeared in AMCAT exam')
          plt.show()
```



## Observations: Most of the people who have appeared for AMCAT-2015 are from Uttar Pradesh

In [387]:

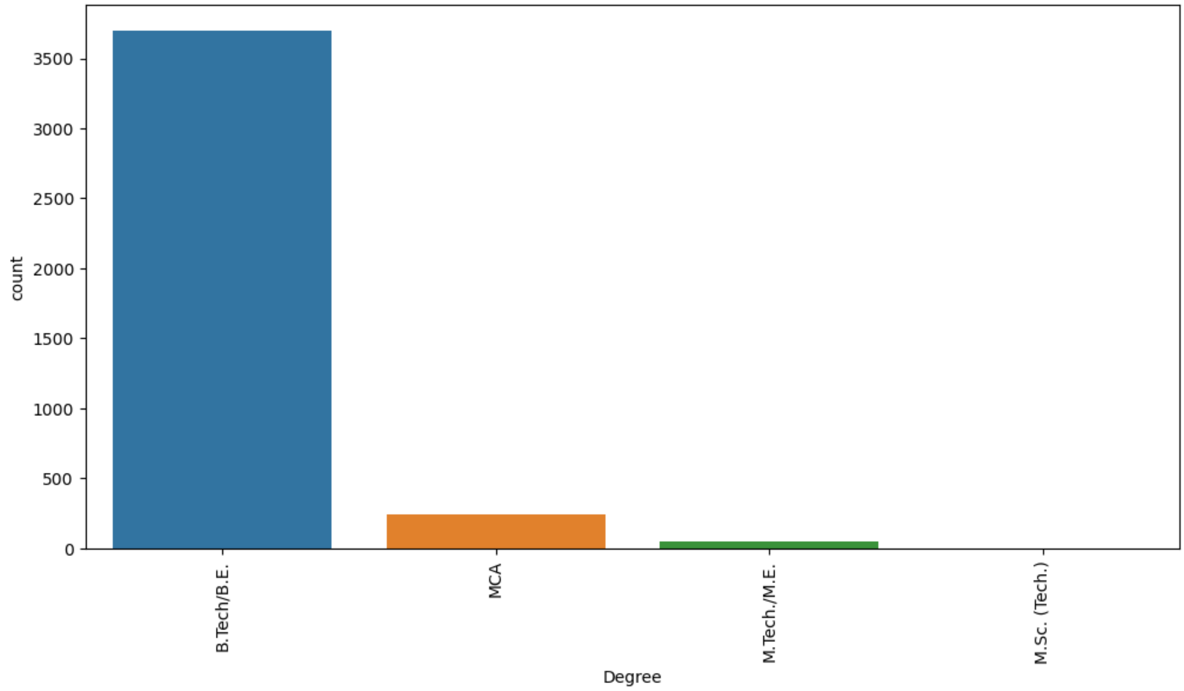
```
plt.figure(figsize=(12, 6))  # Adjust the figure size
sns.countplot(x='Specialization', data=df)
plt.xticks(rotation=90)
plt.show()
```



## Observations: Most of the students are from Computer Science background.

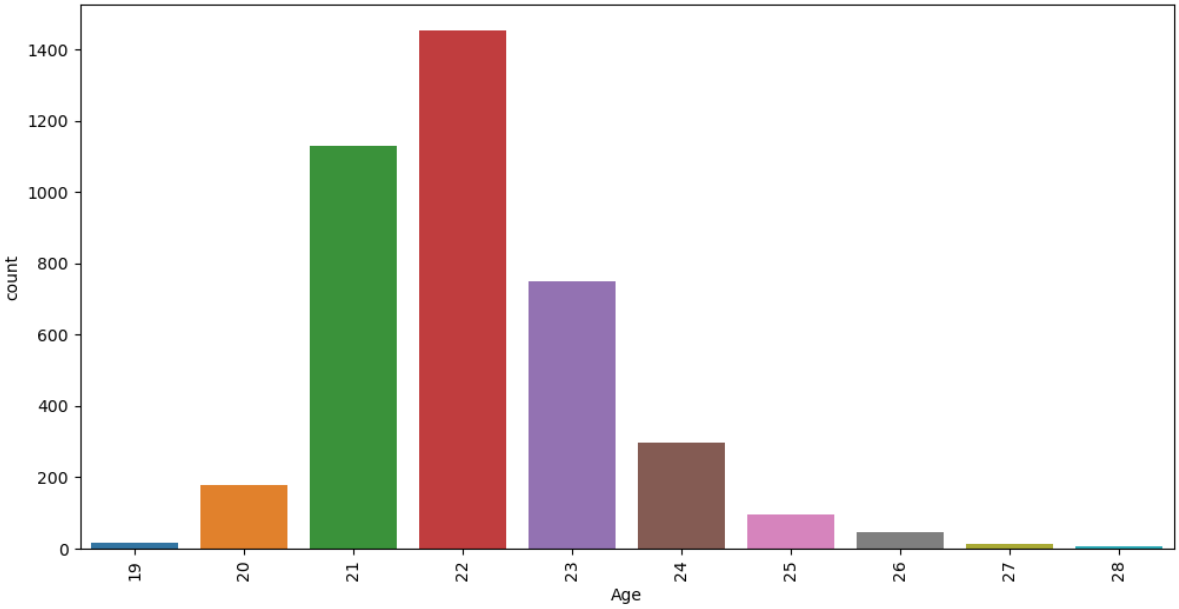
In [132]:

```
plt.figure(figsize=(12, 6))  # Adjust the figure size
sns.countplot(x='Degree', data=df)
plt.xticks(rotation=90)
plt.show()
```



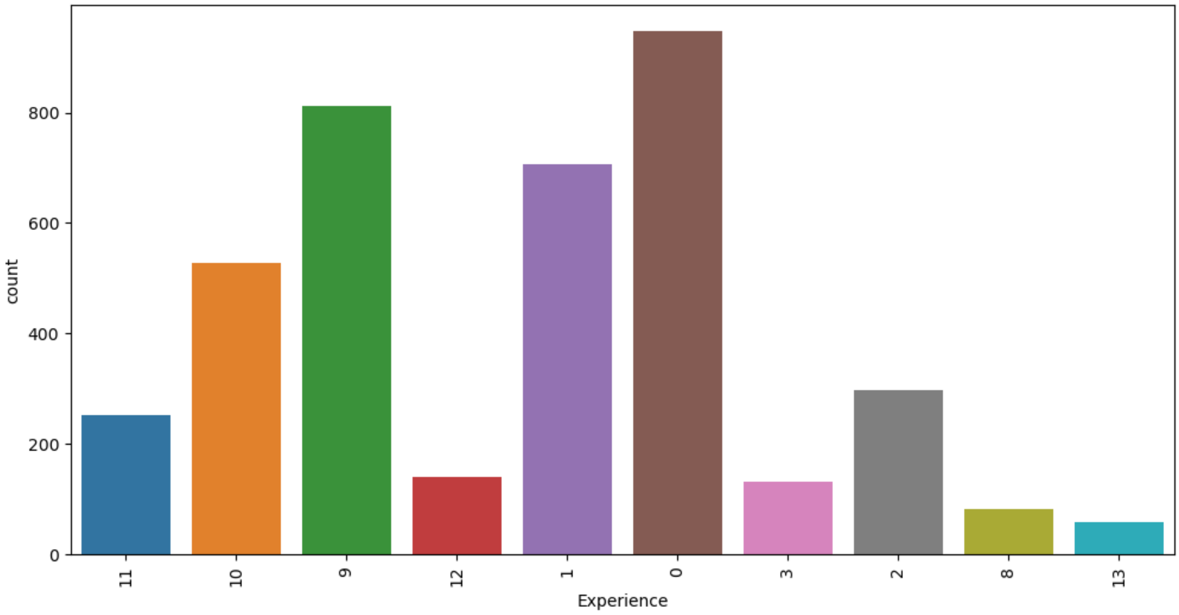
## Observations: B.Tech/B.E students have predominantly appeared for AMCAT exam.

In [152]:
```
top_values = df['Age'].value_counts().nlargest(10).index
plt.figure(figsize=(12, 6))
sns.countplot(x='Age', data=df[df['Age'].isin(top_values)])
plt.xticks(rotation=90)
plt.show()
```



## Observations: Most of the working professionals are of age nearly 22 years.

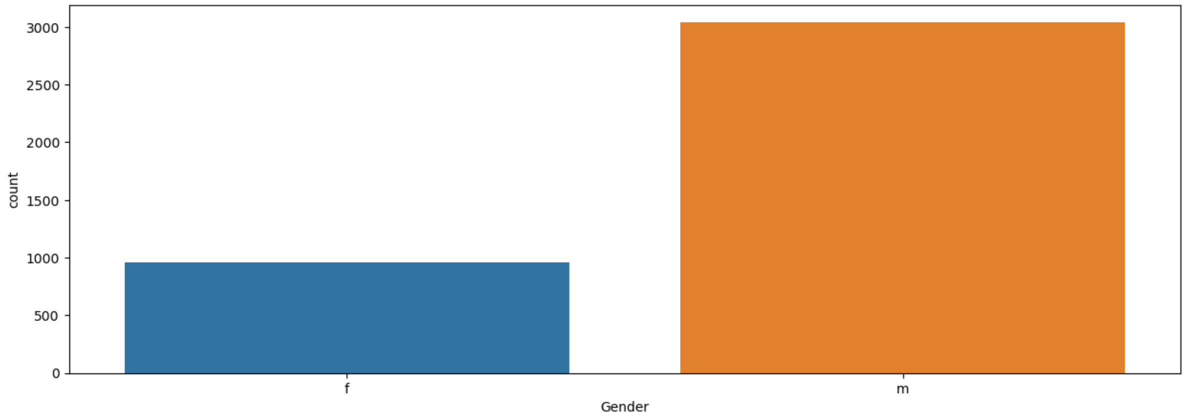
In [157]:
```
top_values = df['Experience'].value_counts().nlargest(10).index
plt.figure(figsize=(12, 6))
sns.countplot(x='Experience', data=df[df['Experience'].isin(top_values)])
plt.xticks(rotation=90)
plt.show()
```



### Observations: From this bar plot we can infer that most of the job-holders are freshers with 0 years of experience.

In [166]:

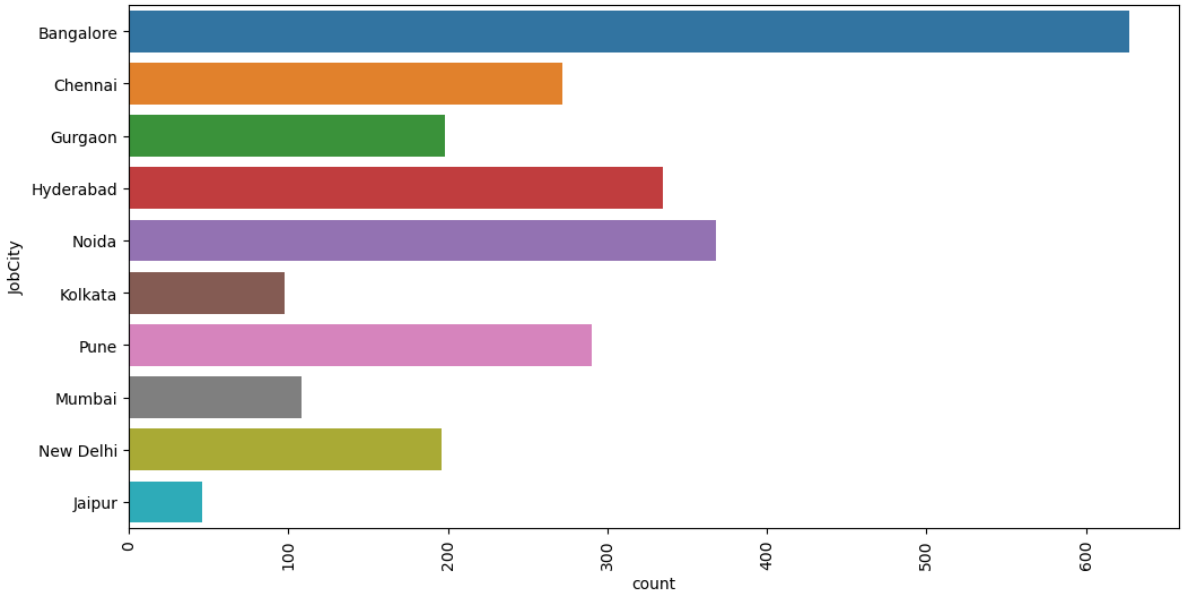
```
plt.figure(figsize=(15, 5))
sns.countplot(data=df, x='Gender')
plt.show()
```



### Observations: Males have majorly appeared for the AMCAT exam.

In [139]:

```
top_values = df.loc[df['JobCity']!='N/A','JobCity'].value_counts().nlargest(10).i
plt.figure(figsize=(12, 6))
sns.countplot(y='JobCity', data=df[df['JobCity'].isin(top_values)])
plt.xticks(rotation=90)
plt.show()
```



### Observations: We can say that most of the job-holders are from Bangalore.

In [ ]:

# Bivariate & Multivariate Analysis

```
In [158]: df.groupby('Gender')['conscientiousness', 'agreeableness', 'extraversion','nuerot
```

```
Out[158]:
                conscientiousness  agreeableness  extraversion  nueroticism  openess_to_experience
        Gender
             f           0.121034       0.292444      0.012173    -0.179358               0.038246
             m          -0.088228       0.100475     -0.000101    -0.165719              -0.193264
```

- Observations: From this we can say that, females are overall having a better personality traits compared to males i.e.,'conscientiousness', 'agreeableness', 'extraversion','nueroticism', 'openess_to_experience'

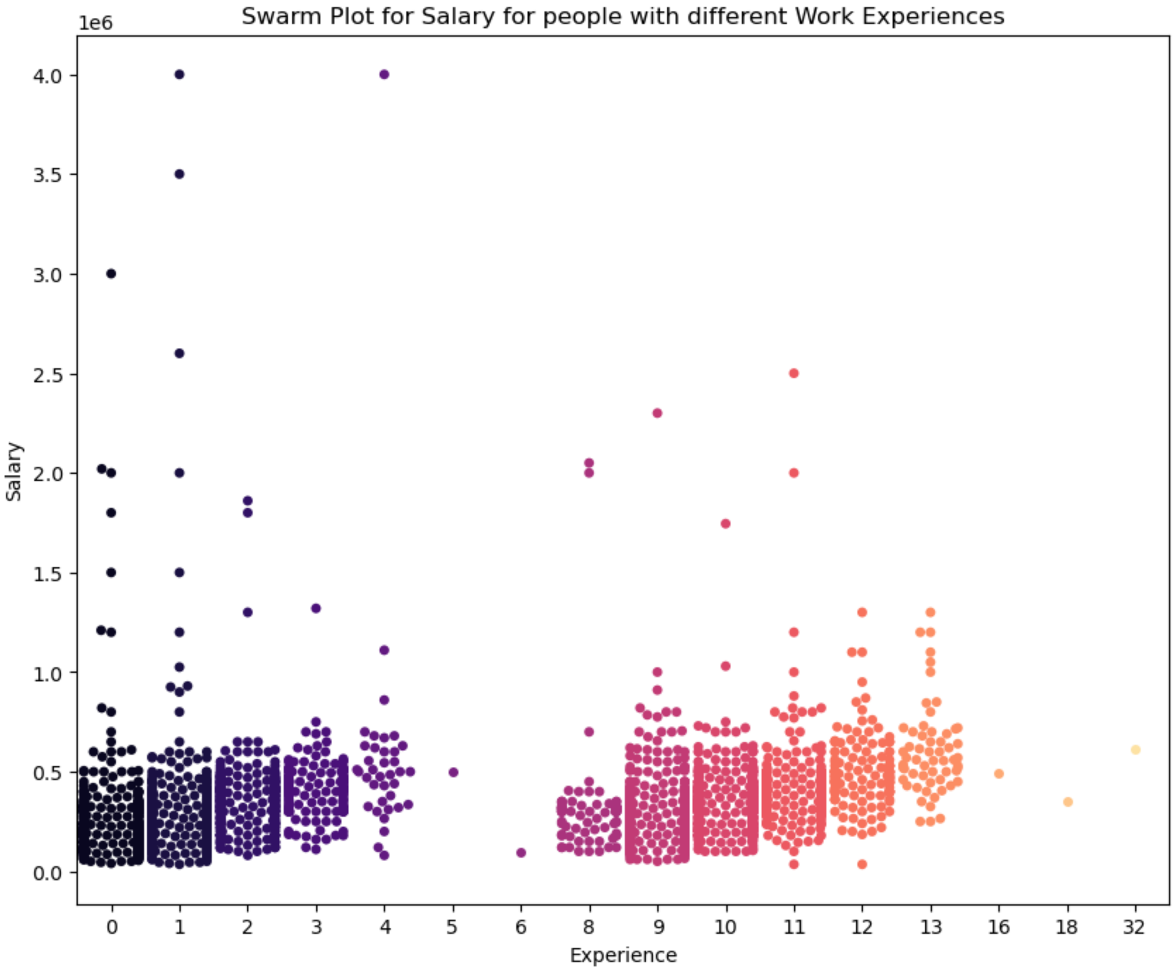
```
In [159]: pd.DataFrame(df.groupby('Gender')['Salary'].mean())
```

```
Out[159]:
                         Salary
        Gender
             f   294937.304075
             m   311711.842105
```

- We can infer from this that Males are having a better salary-pay then females on an average

```
In [ ]:
```

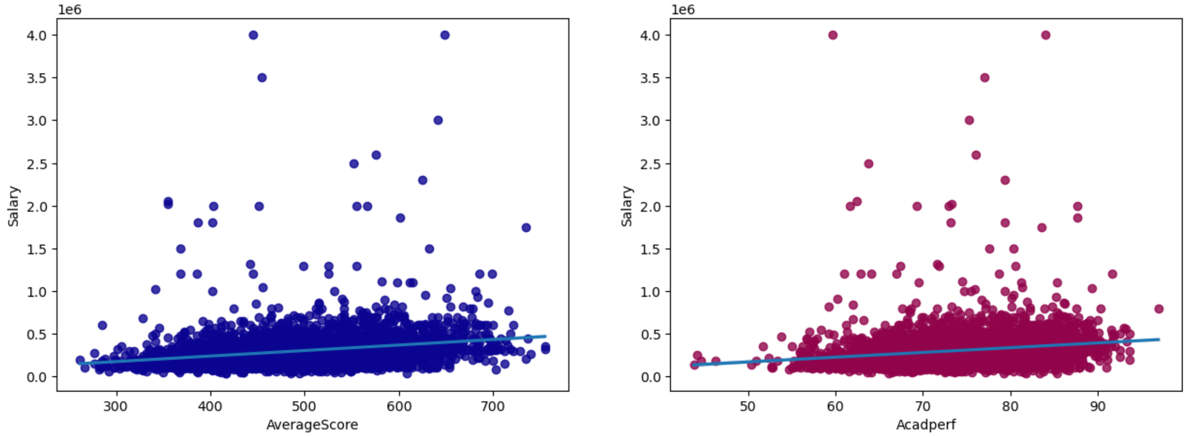
In [326]:
```
plt.figure(figsize=(10, 8))

sns.swarmplot(x="Experience", y="Salary", data=df,palette='magma')

plt.title('Swarm Plot for Salary for people with different Work Experiences')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.show()
```



## Observations: This plot represents the salaries of different experienced people

In [355]:

```
plt.figure(figsize=(15, 5))

# Calculate Average Score and Academic Performance
df['AverageScore'] = (df['Logical'] + df['Quant'] + df['English']) / 3
df['Acadperf'] = (df['10percentage'] + df['12percentage'] + df['collegeGPA']) / 3

# Plotting the regression plots with color
plt.subplot(1, 2, 1)
sns.regplot(x='AverageScore', y='Salary', data=df, scatter_kws={"color": "#0b0491
plt.subplot(1, 2, 2)
sns.regplot(x='Acadperf', y='Salary', data=df, scatter_kws={"color": "#91044b"})
plt.show()
```



## Observation:From this we can say that there is some positive correlation between Average Score and Acadperf

In [ ]:

In [341]:

```
box_city=df[df['JobCity']!='N/A']['JobCity'].value_counts(ascending=False)[0:5].r
```

In [345]:

```
box_city.rename({'index':'JobCity','JobCity':'Count'},inplace=True,axis=1)
```

In [346]:

```
box_city
```

Out[346]:

| | JobCity | Count |
|---|---|---|
| **0** | Bangalore | 685 |
| **1** | Noida | 420 |
| **2** | Hyderabad | 370 |
| **3** | Pune | 328 |
| **4** | Chennai | 313 |

In [348]:

```
df_box=pd.merge(box_city,df,on='JobCity',how='inner')
```

In [351]:

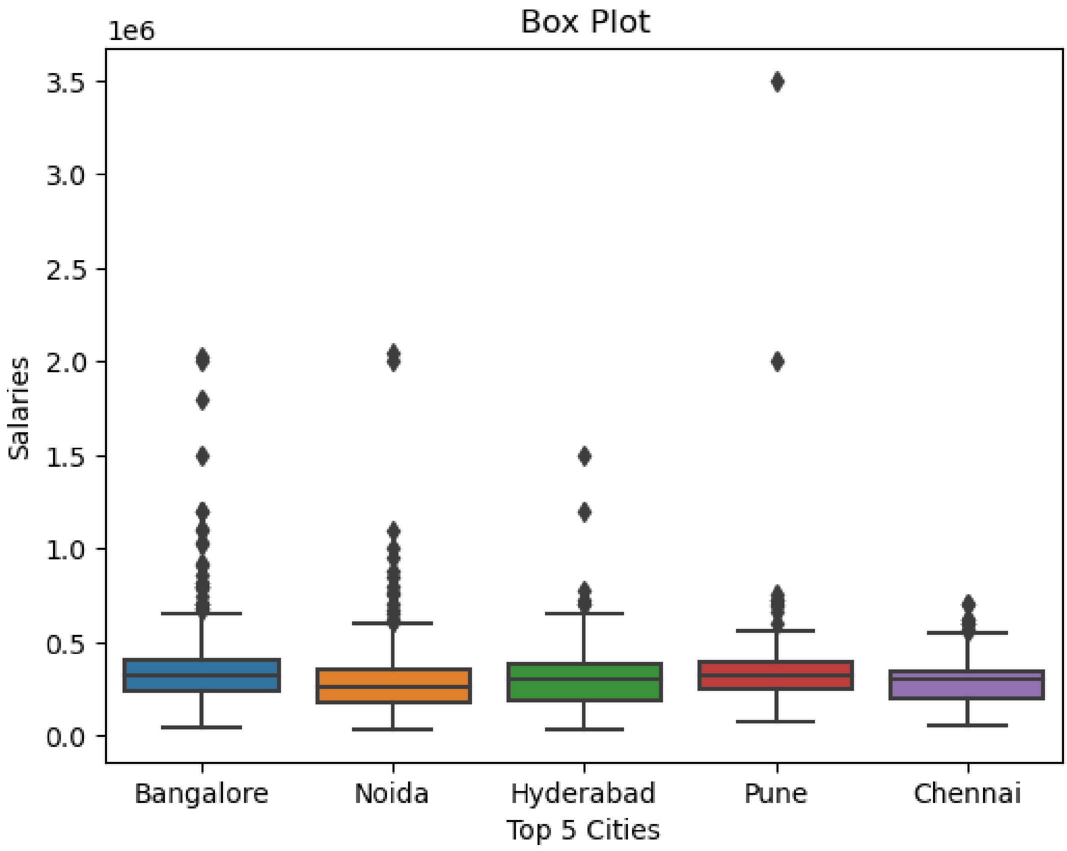
```
df_box.head()
```

Out[351]:

|  | JobCity | Count | ID | Salary | DOJ | DOL | Designation | Gender | DOB | 10percentage | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Bangalore | 685 | 203097 | 420000.0 | 2012-06-01 | 2024-02-23 | senior quality engineer | f | 1990-02-19 | 84.30 | |
| **1** | Bangalore | 685 | 947847 | 300000.0 | 2014-08-01 | 2015-05-01 | java software engineer | m | 1993-02-01 | 86.08 | |
| **2** | Bangalore | 685 | 912934 | 400000.0 | 2014-07-01 | 2015-07-01 | mechanical engineer | m | 1992-05-27 | 92.00 | |
| **3** | Bangalore | 685 | 87291 | 600000.0 | 2011-04-01 | 2015-04-01 | senior php developer | m | 1989-06-24 | 88.60 | |
| **4** | Bangalore | 685 | 1279958 | 300000.0 | 2013-07-01 | 2024-02-23 | java software engineer | m | 1992-07-02 | 81.20 | |

In [352]:

```
sns.boxplot(x='JobCity', y='Salary', data=df_box)
plt.xlabel('Top 5 Cities')
plt.ylabel('Salaries')
plt.title('Box Plot')
plt.show()
```

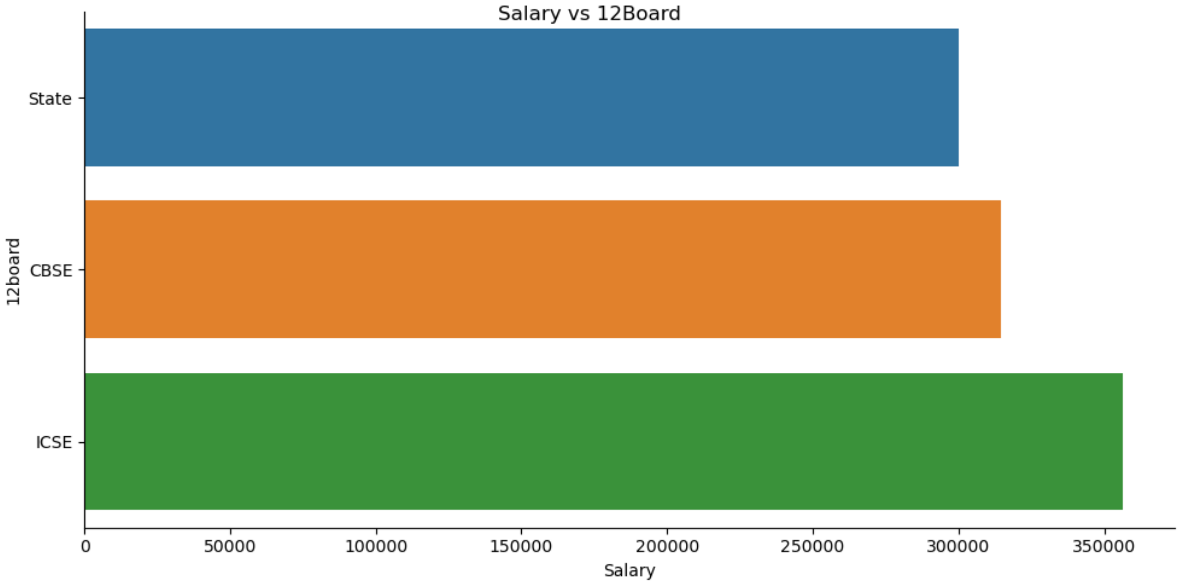


## Observations: Top 5 Cities(based on workforce) and their salaries

In [ ]:

In [165]:

```
sns.catplot(x='Salary', y='12board', kind='bar', data=df, ci=None, height=5, aspe
plt.suptitle('Salary vs 12Board')
plt.show()
```

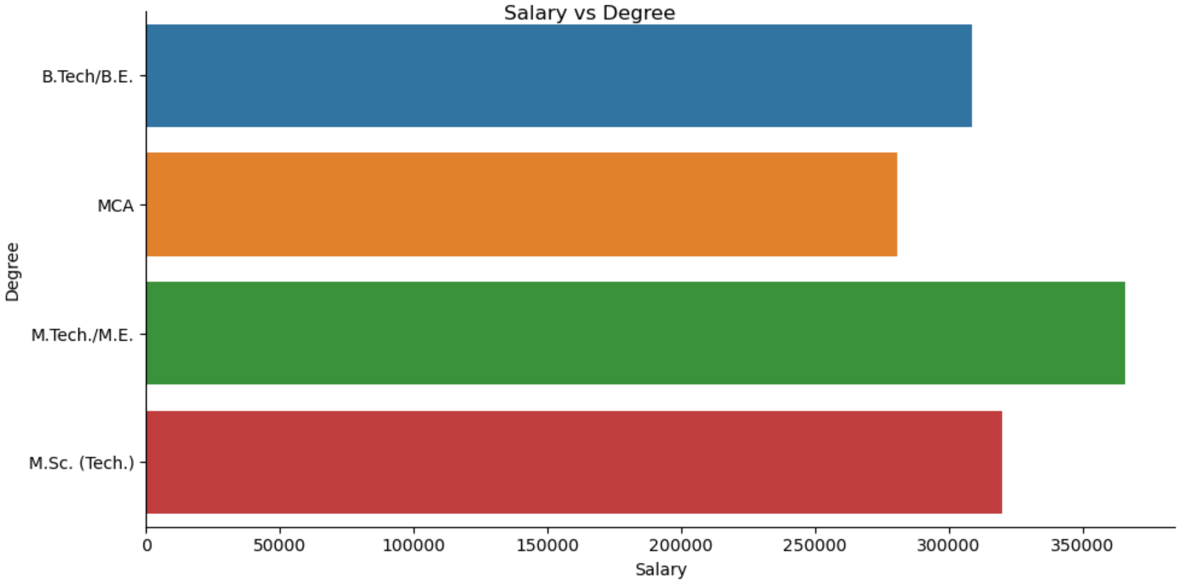


### Observations: From this plot we can say that students who have passed out of 12th board of ICSE

In [ ]:

In [61]:

```
sns.catplot(x='Salary', y='Degree', kind='bar', data=df, ci=None, height=5, aspec
plt.suptitle('Salary vs Degree')
plt.show()
```



### Observations: We can observe that, M.Tech/M.E graduates have a higher salary

In [ ]:

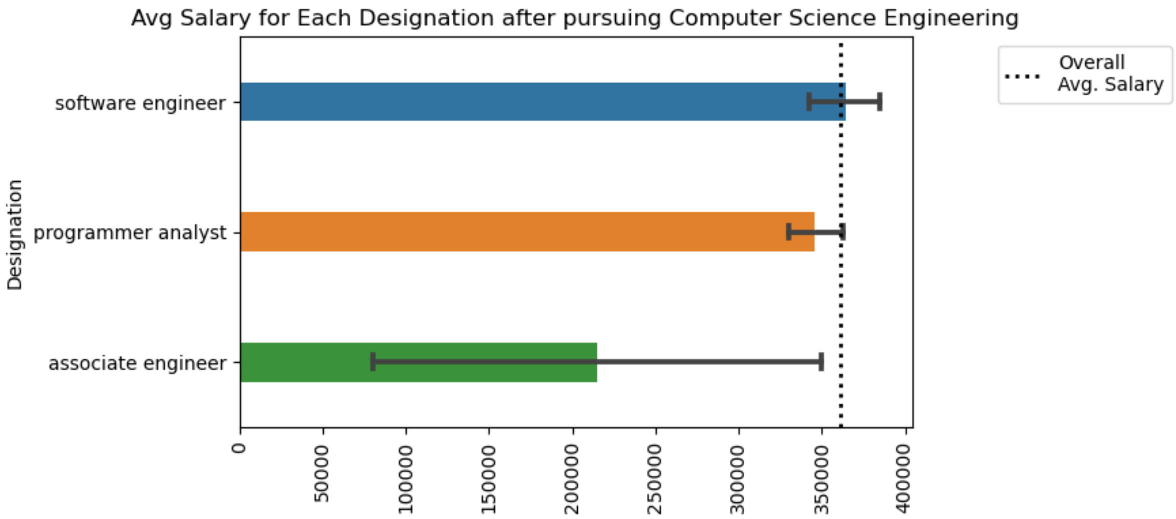
# Research Questions

### 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 [369]:
```
df['Designation'] = df['Designation'].replace([
    'programmer analyst trainee', 'programmer analyst'
], 'programmer analyst'
)

df['Designation'] = df['Designation'].replace([
    'software eng', 'software engg', 'software engineer', 'software engineere', '
], 'software engineer'
)
```

In [370]:
```
df2 = df[(df["Designation"].isin(["programmer analyst", "software engineer", "har
                (df["Specialization"].isin(["CSE"])) & (df['DOJ'].dt.year==df['Gr
```

In [371]:
```
fig, ax = plt.subplots(figsize=(10, 4))
sns.barplot(x='Salary', y='Designation',
            data=df2,
            capsize=0.1,
            width=0.3,
            ax=ax)
ax.axvline(df2['Salary'].mean(), color='k',
           linestyle=':',
           linewidth=2, label='Overall\nAvg. Salary')
ax.set_title('Avg Salary for Each Designation after pursuing Computer Science Eng
ax.legend(loc='upper right', bbox_to_anchor=(1.4, 1))
ax.set_xlabel('')
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)

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



In [372]:
```
df2['Salary'].nunique()
```

Out[372]:
```
83
```

In [373]:
```
from scipy import stats as st
popmean = 250000 + 300000 / 2
pv = st.ttest_1samp(df2['Salary'], popmean=popmean)[1]
alpha = 0.05
if pv < alpha:
    print('We reject the null hypothesis and Average salary is not equal to 250k'
else:
    print('We fail to reject null hypothesis and Average salary is equal to 250k'
```

```
We reject the null hypothesis and Average salary is not equal to 250k
```

### Therefore we can say that the claim Times of India making is not correct

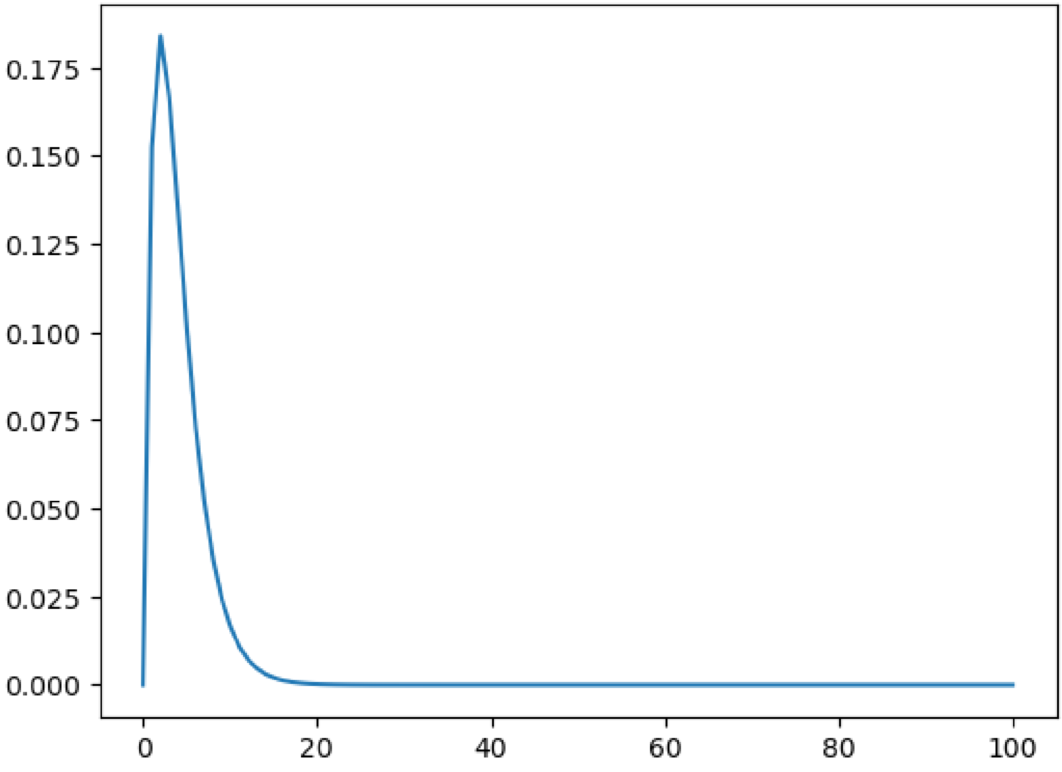
In [ ]:
```

```

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

In [374]:
```
from scipy.stats import chi2
from scipy.stats import chi2_contingency
```

In [375]:
```
x = np.linspace(0, 100, 100)
y = chi2.pdf(x, df = 4)
plt.plot(x, y)
```

Out[375]:
```
[<matplotlib.lines.Line2D at 0x213616df210>]
```



In [376]:

```
obsr = pd.crosstab(df.Specialization,df.Gender)
obsr
```

Out[376]:

| Gender | f | m |
|---|---|---|
| **Specialization** | | |
| **CE** | 6 | 23 |
| **CSE** | 601 | 1688 |
| **EC** | 306 | 1013 |
| **EL** | 17 | 68 |
| **ME** | 12 | 207 |
| **other** | 15 | 41 |

In [377]:

```
chi2_statistic, chi2_p_value, chi2_dof, chi2_expected = chi2_contingency(obsr)

print("Statistic            :", chi2_statistic)
print('')
print("p value              :", chi2_p_value)
print('')
print("Degrees of freedom   :", chi2_dof)
print('')
print("Expected frequencies array:\n", chi2_expected)
```

```
Statistic             : 49.26560031142505

p value               : 1.9584544175343366e-09

Degrees of freedom    : 5

Expected frequencies array:
 [[   6.94345759   22.05654241]
 [ 548.05429072 1740.94570928]
 [ 315.8076057  1003.1923943 ]
 [  20.35151364   64.64848636]
 [  52.43507631  166.56492369]
 [  13.40805604   42.59194396]]
```

In [378]:

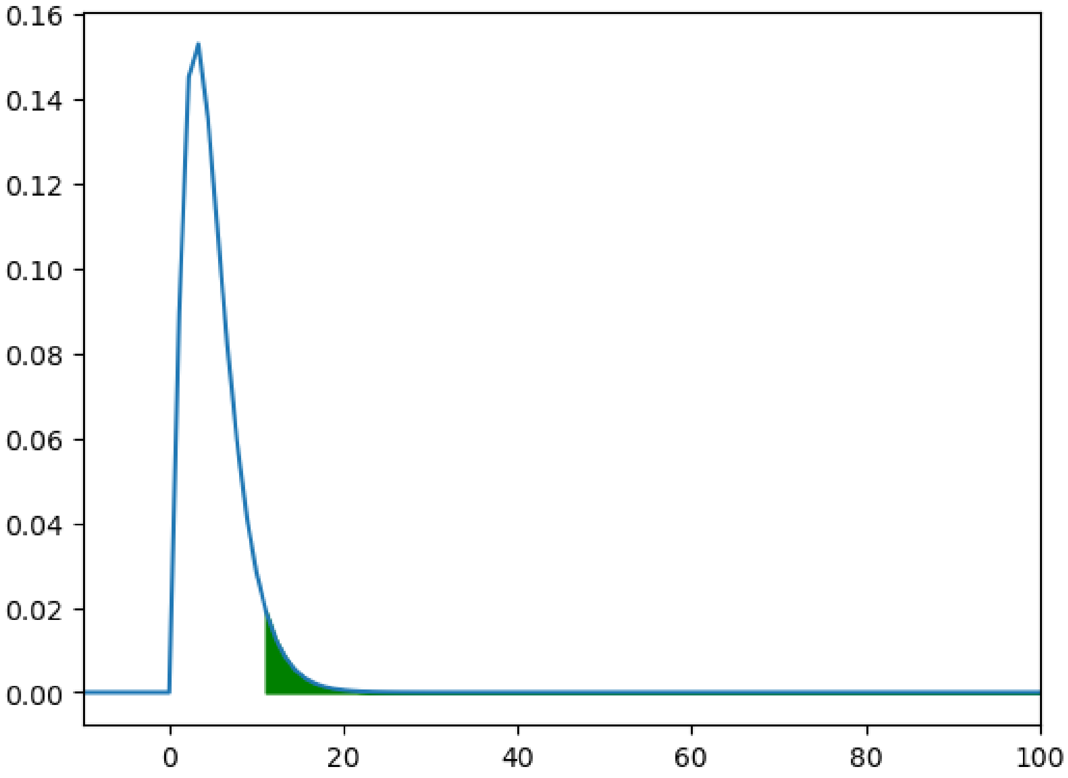
```
confidence_level = 0.95
alpha = 1 - confidence_level

chi2_critical = chi2.ppf(1 - alpha, chi2_dof)

chi2_critical
```

Out[378]:

```
11.070497693516351
```

In [379]:
```
x_min = -10
x_max = 100

x = np.linspace(x_min, x_max, 100)
y = chi2.pdf(x, chi2_dof)
plt.xlim(x_min, x_max)
plt.plot(x, y)

chi2_critical_right = chi2_critical

x1 = np.linspace(chi2_critical_right, x_max, 100)
y1 = chi2.pdf(x1, chi2_dof)
plt.fill_between(x1, y1, color='green')
```

Out[379]:
```
<matplotlib.collections.PolyCollection at 0x21361a482d0>
```



In [380]:
```
if(chi2_statistic > chi2_critical):
    print("There is not enough evidence to reject the Null Hypothesis")
else:
    print("There is sufficent evidence to reject the Null Hypothesis")
```

```
There is not enough evidence to reject the Null Hypothesis
```

In [381]:
```
if(chi2_p_value < alpha):
    print("There is not enough evidence to reject the Null Hypothesis")
else:
    print("There is sufficent evidence to reject the Null Hypothesis")
```

```
There is not enough evidence to reject the Null Hypothesis
```

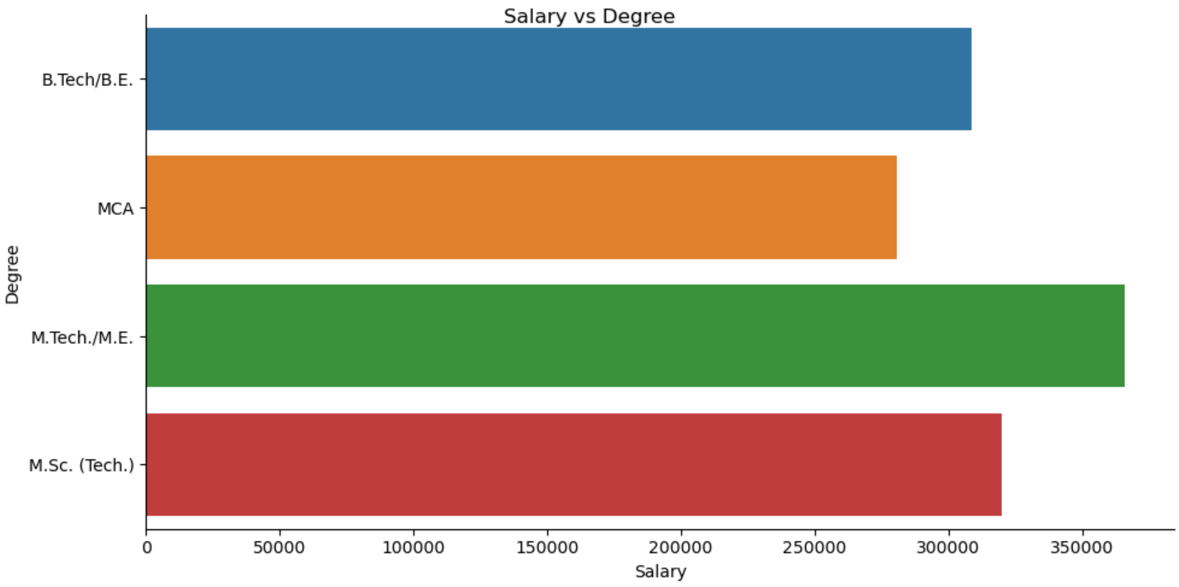
### There is no enough evidence to say that Gender and specialization are related

## Creating a research question

### Is there any relation between the Degree and Salary

In [386]:

```
sns.catplot(x='Salary', y='Degree', kind='bar', data=df, ci=None, height=5, aspec
plt.suptitle('Salary vs Degree')
plt.show()
```



### We can say from this plot that M.Tech/M.E students had a higher pay

## Conclusion

#### Data Understanding:

The dataset encompasses the employment outcomes of engineering graduates, focusing on target variable Salary. Additionally, it includes standardized scores in three distinct areas: cognitive skills, technical skills, and personality skills.

#### Data Manipulation:

Upon initial observation, the dataset consists of 4000 rows and 40 columns. The dataset exhibits numerous duplicate values, necessitating data manipulation. Initially, we remove redundant rows and columns. Subsequently, we assess for the presence of any missing values (NaN). Following data cleaning, we proceed with visualization. Data Visualization:

#### Univariate Analysis:

Univariate analysis encompasses various plots, including Cumulativee Distribution Functions (CDF), Histograms, Box Plots, and Summary Plots. These visualizations illustrate probability and frequency distributions.

**Bivariate Analysis:**

Bivariate analysis comprises Scatterplots, Barplots, Crosstabs, Pivot tables, pie charts. This analysis helps in comparing percentages across different variables. Additionally, it aids in identifying

In [ ]: