amcat-analysis

February 18, 2024

1 ANALYSIS OF AMCAT DATA - EDA

1.0.1 The AMCAT test, or the Aspiring Minds Computer Adaptive Test, is a career aptitude exam for jobseekers in India.

1.0.2 ASPIRING MINDS FROM ASPIRING MINDS EMPLOYMENT OUTCOME 2015 (AMEO)

2 1. ABOUT DATASET

The dataset was released by Aspiring Minds from the Aspiring Mind Employment Outcome 2015 (AMEO). The study is primarily limited only to students with engineering disciplines. The dataset contains the employment outcomes of engineering graduates as dependent variables (Salary, Job Titles, and Job Locations) along with the standardized scores from three different areas – cognitive skills, technical skills and personality skills. The dataset also contains demographic features. The dataset contains around 40 independent variables and 4000 data points. The independent variables are both continuous and categorical in nature. The dataset contains a unique identifier for each candidate. Below mentioned table contains the details for the original dataset.

[29]:
```
# Importing Libraries

import pandas as pandu
import matplotlib.pyplot as pluto
import seaborn as sobu
```

[28]:
```
# Importing Dataset

dataframe = pandu.read_excel("/content/data.xlsx")
```

3 2. DATA EXPLORATION

[5]:
```
print(dataframe.head(10))
```

```
  Unnamed: 0      ID   Salary        DOJ                   DOL  \
0      train  203097   420000 2012-06-01               present
1      train  579905   500000 2013-09-01               present
2      train  810601   325000 2014-06-01               present
3      train  267447  1100000 2011-07-01               present
```

```
4      train   343523   200000 2014-03-01  2015-03-01 00:00:00
5      train  1027655   300000 2014-06-01              present
6      train   947847   300000 2014-08-01  2015-05-01 00:00:00
7      train   912934   400000 2014-07-01  2015-07-01 00:00:00
8      train   552574   600000 2013-07-01              present
9      train  1203363   230000 2014-07-01              present

                Designation    JobCity Gender        DOB  10percentage  …  \
0   senior quality engineer  Bangalore      f 1990-02-19         84.30  …
1         assistant manager     Indore      m 1989-10-04         85.40  …
2          systems engineer    Chennai      f 1992-08-03         85.00  …
3  senior software engineer    Gurgaon      m 1989-12-05         85.60  …
4                       get    Manesar      m 1991-02-27         78.00  …
5           system engineer  Hyderabad      m 1992-07-02         89.92  …
6    java software engineer   Banglore      m 1993-02-01         86.08  …
7       mechanical engineer  Bangalore      m 1992-05-27         92.00  …
8       electrical engineer      Noida      m 1991-09-17         90.00  …
9          project engineer    Kolkata      m 1993-06-13         77.00  …

  ComputerScience  MechanicalEngg  ElectricalEngg TelecomEngg  CivilEngg  \
0              -1              -1              -1          -1         -1
1              -1              -1              -1          -1         -1
2              -1              -1              -1          -1         -1
3              -1              -1              -1          -1         -1
4              -1              -1              -1          -1         -1
5             407              -1              -1          -1         -1
6             346              -1              -1          -1         -1
7              -1             469              -1          -1         -1
8              -1              -1              -1          -1         -1
9              -1              -1              -1          -1         -1

   conscientiousness agreeableness extraversion  nueroticism  \
0             0.9737        0.8128       0.5269      1.35490
1            -0.7335        0.3789       1.2396     -0.10760
2             0.2718        1.7109       0.1637     -0.86820
3             0.0464        0.3448      -0.3440     -0.40780
4            -0.8810       -0.2793      -1.0697      0.09163
5            -0.3027       -0.6201      -2.2954     -0.74150
6             1.7081       -0.1054      -1.0379     -2.00920
7            -0.0154        1.2114       0.0100      0.14590
8            -0.1590        0.5454      -0.6048     -0.74150
9            -1.3080        0.5454      -0.9122      0.90660

   openess_to_experience
0                -0.4455
1                 0.8637
2                 0.6721
3                -0.9194
```

```
4                -0.1295
5                -0.8608
6                -1.0872
7                 1.2470
8                -0.2859
9                 0.0973

[10 rows x 39 columns]
```

[6]:
```
print(dataframe.shape)
```

```
(3998, 39)
```

[7]:
```
print(dataframe.describe())
```

```
                 ID        Salary  10percentage  12graduation  12percentage  \
count  3.998000e+03  3.998000e+03   3998.000000   3998.000000   3998.000000
mean   6.637945e+05  3.076998e+05     77.925443   2008.087544     74.466366
std    3.632182e+05  2.127375e+05      9.850162      1.653599     10.999933
min    1.124400e+04  3.500000e+04     43.000000   1995.000000     40.000000
25%    3.342842e+05  1.800000e+05     71.680000   2007.000000     66.000000
50%    6.396000e+05  3.000000e+05     79.150000   2008.000000     74.400000
75%    9.904800e+05  3.700000e+05     85.670000   2009.000000     82.600000
max    1.298275e+06  4.000000e+06     97.760000   2013.000000     98.700000

          CollegeID  CollegeTier   collegeGPA  CollegeCityID  CollegeCityTier  \
count   3998.000000  3998.000000  3998.000000    3998.000000      3998.000000
mean    5156.851426     1.925713    71.486171    5156.851426         0.300400
std     4802.261482     0.262270     8.167338    4802.261482         0.458489
min        2.000000     1.000000     6.450000       2.000000         0.000000
25%      494.000000     2.000000    66.407500     494.000000         0.000000
50%     3879.000000     2.000000    71.720000    3879.000000         0.000000
75%     8818.000000     2.000000    76.327500    8818.000000         1.000000
max    18409.000000     2.000000    99.930000   18409.000000         1.000000

       …  ComputerScience  MechanicalEngg  ElectricalEngg  TelecomEngg  \
count  …      3998.000000     3998.000000     3998.000000  3998.000000
mean   …        90.742371       22.974737       16.478739    31.851176
std    …       175.273083       98.123311       87.585634   104.852845
min    …        -1.000000       -1.000000       -1.000000    -1.000000
25%    …        -1.000000       -1.000000       -1.000000    -1.000000
50%    …        -1.000000       -1.000000       -1.000000    -1.000000
75%    …        -1.000000       -1.000000       -1.000000    -1.000000
max    …       715.000000      623.000000      676.000000   548.000000

         CivilEngg  conscientiousness  agreeableness  extraversion  \
count  3998.000000        3998.000000    3998.000000   3998.000000
mean      2.683842          -0.037831       0.146496      0.002763
```

```
std      36.658505            1.028666      0.941782      0.951471
min      -1.000000           -4.126700     -5.781600     -4.600900
25%      -1.000000           -0.713525     -0.287100     -0.604800
50%      -1.000000            0.046400      0.212400      0.091400
75%      -1.000000            0.702700      0.812800      0.672000
max     516.000000            1.995300      1.904800      2.535400

       nueroticism  openess_to_experience
count  3998.000000            3998.000000
mean     -0.169033              -0.138110
std       1.007580               1.008075
min      -2.643000              -7.375700
25%      -0.868200              -0.669200
50%      -0.234400              -0.094300
75%       0.526200               0.502400
max       3.352500               1.822400

[8 rows x 27 columns]
```

[8]:
```python
print(dataframe.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             3998 non-null   object
 1   ID                     3998 non-null   int64
 2   Salary                 3998 non-null   int64
 3   DOJ                    3998 non-null   datetime64[ns]
 4   DOL                    3998 non-null   object
 5   Designation            3998 non-null   object
 6   JobCity                3998 non-null   object
 7   Gender                 3998 non-null   object
 8   DOB                    3998 non-null   datetime64[ns]
 9   10percentage           3998 non-null   float64
 10  10board                3998 non-null   object
 11  12graduation           3998 non-null   int64
 12  12percentage           3998 non-null   float64
 13  12board                3998 non-null   object
 14  CollegeID              3998 non-null   int64
 15  CollegeTier            3998 non-null   int64
 16  Degree                 3998 non-null   object
 17  Specialization         3998 non-null   object
 18  collegeGPA             3998 non-null   float64
 19  CollegeCityID          3998 non-null   int64
 20  CollegeCityTier        3998 non-null   int64
 21  CollegeState           3998 non-null   object
```

```
 22  GraduationYear         3998 non-null   int64
 23  English                3998 non-null   int64
 24  Logical                3998 non-null   int64
 25  Quant                  3998 non-null   int64
 26  Domain                 3998 non-null   float64
 27  ComputerProgramming    3998 non-null   int64
 28  ElectronicsAndSemicon  3998 non-null   int64
 29  ComputerScience        3998 non-null   int64
 30  MechanicalEngg         3998 non-null   int64
 31  ElectricalEngg         3998 non-null   int64
 32  TelecomEngg            3998 non-null   int64
 33  CivilEngg              3998 non-null   int64
 34  conscientiousness      3998 non-null   float64
 35  agreeableness          3998 non-null   float64
 36  extraversion           3998 non-null   float64
 37  nueroticism            3998 non-null   float64
 38  openess_to_experience  3998 non-null   float64
dtypes: datetime64[ns](2), float64(9), int64(18), object(10)
memory usage: 1.2+ MB
None
```

[52]:

```
# Missing Values Handling

MV = dataframe.isnull().sum()
print("Missing Values:-\n", MV)
```

```
Missing Values:-
 Unnamed: 0               0
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
Age                      0
dtype: int64
```

[53]:
```
# Handling Duplicate Rows

DR = dataframe.duplicated().sum()
print("Duplicate Rows:-", DR)
dataframe.drop_duplicates(inplace=True)
```

```
Duplicate Rows:- 0
```

4 3. UNIVARIATE ANALYSIS

[14]:
```
from matplotlib import colormaps
list(colormaps)
```

[14]:
```
['magma',
 'inferno',
 'plasma',
 'viridis',
 'cividis',
 'twilight',
 'twilight_shifted',
 'turbo',
 'Blues',
 'BrBG',
 'BuGn',
 'BuPu',
 'CMRmap',
```

```
'GnBu',
'Greens',
'Greys',
'OrRd',
'Oranges',
'PRGn',
'PiYG',
'PuBu',
'PuBuGn',
'PuOr',
'PuRd',
'Purples',
'RdBu',
'RdGy',
'RdPu',
'RdYlBu',
'RdYlGn',
'Reds',
'Spectral',
'Wistia',
'YlGn',
'YlGnBu',
'YlOrBr',
'YlOrRd',
'afmhot',
'autumn',
'binary',
'bone',
'brg',
'bwr',
'cool',
'coolwarm',
'copper',
'cubehelix',
'flag',
'gist_earth',
'gist_gray',
'gist_heat',
'gist_ncar',
'gist_rainbow',
'gist_stern',
'gist_yarg',
'gnuplot',
'gnuplot2',
'gray',
'hot',
'hsv',
```

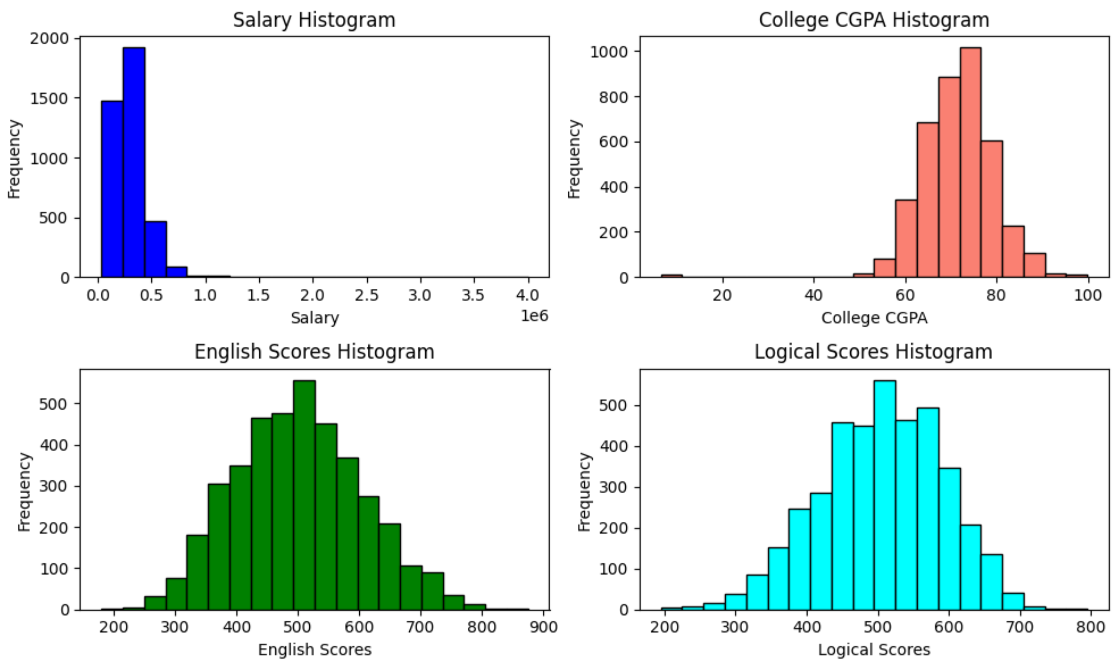
```
'jet',
'nipy_spectral',
'ocean',
'pink',
'prism',
'rainbow',
'seismic',
'spring',
'summer',
'terrain',
'winter',
'Accent',
'Dark2',
'Paired',
'Pastel1',
'Pastel2',
'Set1',
'Set2',
'Set3',
'tab10',
'tab20',
'tab20b',
'tab20c',
'magma_r',
'inferno_r',
'plasma_r',
'viridis_r',
'cividis_r',
'twilight_r',
'twilight_shifted_r',
'turbo_r',
'Blues_r',
'BrBG_r',
'BuGn_r',
'BuPu_r',
'CMRmap_r',
'GnBu_r',
'Greens_r',
'Greys_r',
'OrRd_r',
'Oranges_r',
'PRGn_r',
'PiYG_r',
'PuBu_r',
'PuBuGn_r',
'PuOr_r',
'PuRd_r',
```

```
'Purples_r',
'RdBu_r',
'RdGy_r',
'RdPu_r',
'RdYlBu_r',
'RdYlGn_r',
'Reds_r',
'Spectral_r',
'Wistia_r',
'YlGn_r',
'YlGnBu_r',
'YlOrBr_r',
'YlOrRd_r',
'afmhot_r',
'autumn_r',
'binary_r',
'bone_r',
'brg_r',
'bwr_r',
'cool_r',
'coolwarm_r',
'copper_r',
'cubehelix_r',
'flag_r',
'gist_earth_r',
'gist_gray_r',
'gist_heat_r',
'gist_ncar_r',
'gist_rainbow_r',
'gist_stern_r',
'gist_yarg_r',
'gnuplot_r',
'gnuplot2_r',
'gray_r',
'hot_r',
'hsv_r',
'jet_r',
'nipy_spectral_r',
'ocean_r',
'pink_r',
'prism_r',
'rainbow_r',
'seismic_r',
'spring_r',
'summer_r',
'terrain_r',
'winter_r',
```

```
 'Accent_r',
 'Dark2_r',
 'Paired_r',
 'Pastel1_r',
 'Pastel2_r',
 'Set1_r',
 'Set2_r',
 'Set3_r',
 'tab10_r',
 'tab20_r',
 'tab20b_r',
 'tab20c_r',
 'rocket',
 'rocket_r',
 'mako',
 'mako_r',
 'icefire',
 'icefire_r',
 'vlag',
 'vlag_r',
 'flare',
 'flare_r',
 'crest',
 'crest_r']
```

[74]:
```
# Histogram Graph

pluto.figure(figsize=(10, 6))
pluto.subplot(2, 2, 1)
pluto.hist(dataframe['Salary'], bins=20, color='blue', edgecolor='black')
pluto.title('Salary Histogram')
pluto.xlabel('Salary')
pluto.ylabel('Frequency')

pluto.subplot(2, 2, 2)
pluto.hist(dataframe['collegeGPA'], bins=20, color='salmon', edgecolor='black')
pluto.title('College CGPA Histogram')
pluto.xlabel('College CGPA')
pluto.ylabel('Frequency')

pluto.subplot(2, 2, 3)
pluto.hist(dataframe['English'], bins=20, color='green', edgecolor='black')
pluto.title('English Scores Histogram')
pluto.xlabel('English Scores')
pluto.ylabel('Frequency')

pluto.subplot(2, 2, 4)
```

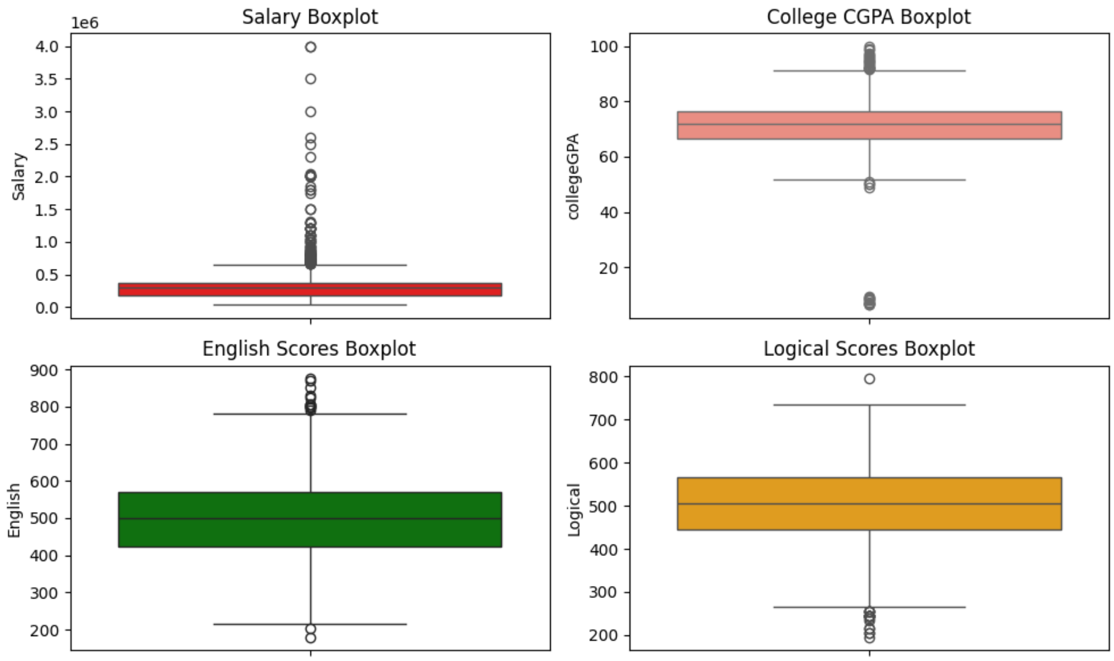
```
pluto.hist(dataframe['Logical'], bins=20, color='cyan', edgecolor='black')
pluto.title('Logical Scores Histogram')
pluto.xlabel('Logical Scores')
pluto.ylabel('Frequency')

pluto.tight_layout()
pluto.show()
```



[75]:

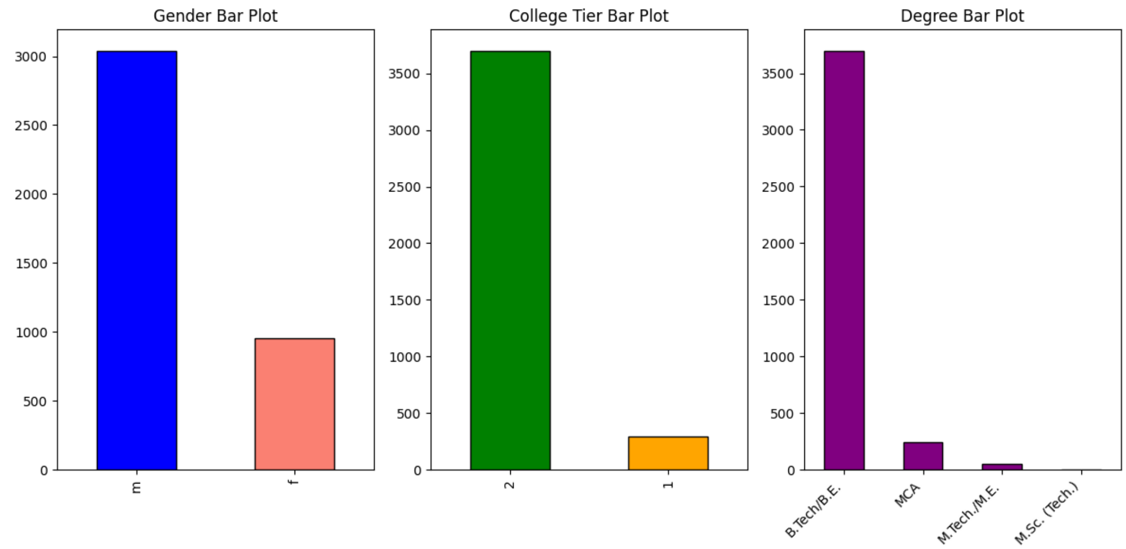
```
# Boxplot Graph

pluto.figure(figsize=(10, 6))
pluto.subplot(2, 2, 1)
sobu.boxplot(dataframe['Salary'], color='Red')
pluto.title('Salary Boxplot')

pluto.subplot(2, 2, 2)
sobu.boxplot(dataframe['collegeGPA'], color='salmon')
pluto.title('College CGPA Boxplot')

pluto.subplot(2, 2, 3)
sobu.boxplot(dataframe['English'], color='green')
pluto.title('English Scores Boxplot')

pluto.subplot(2, 2, 4)
sobu.boxplot(dataframe['Logical'], color='orange')
```

```
pluto.title('Logical Scores Boxplot')

pluto.tight_layout()
pluto.show()
```



[60]:
```
# Categorical Data (BP)

pluto.figure(figsize=(12, 6))
pluto.subplot(1, 3, 1)
dataframe['Gender'].value_counts().plot(kind='bar', color=['blue', 'salmon'],␣
  ↪edgecolor='black')
pluto.title('Gender Bar Plot')

pluto.subplot(1, 3, 2)
dataframe['CollegeTier'].value_counts().plot(kind='bar', color=['green',␣
  ↪'orange'], edgecolor='black')
pluto.title('College Tier Bar Plot')

pluto.subplot(1, 3, 3)
dataframe['Degree'].value_counts().plot(kind='bar', color='purple',␣
  ↪edgecolor='black')
pluto.title('Degree Bar Plot')

pluto.xticks(rotation=45, ha='right')
pluto.tight_layout()
```

```
pluto.show()
```

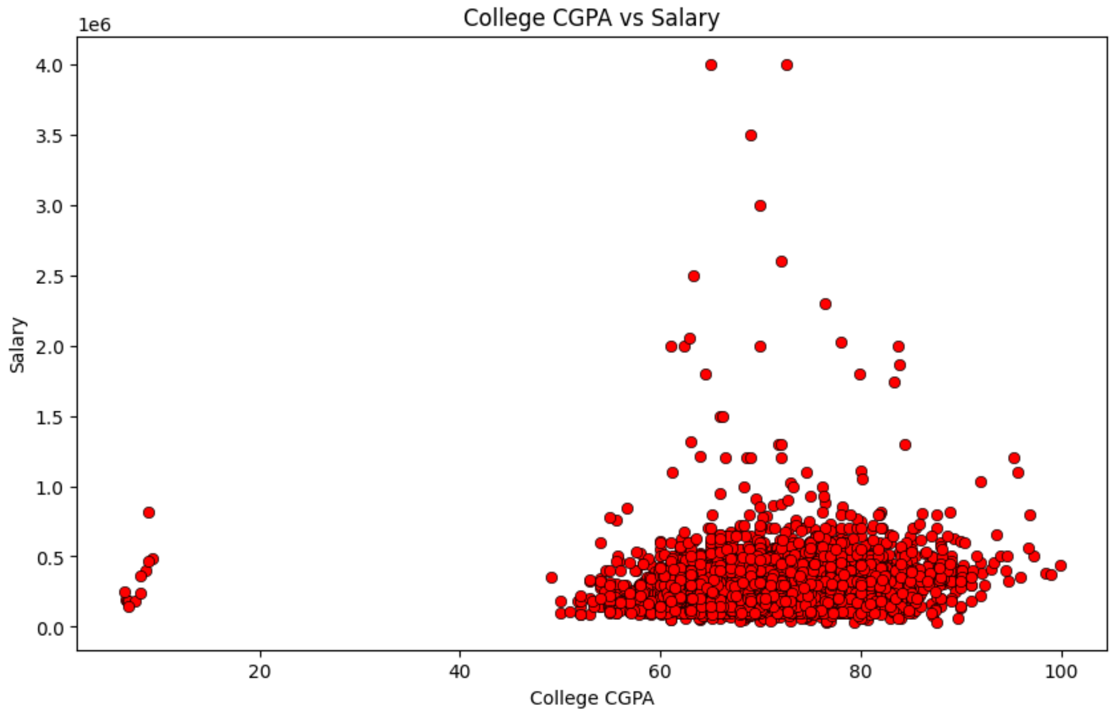


1. **Histograms:-**
   - **Pagar -** Right-skewed distribution with most prices concentrated towards the lower end.
   - **College CGPA -** The normal distribution is almost based on the mean GPA.
   - **English scores -** Coarse normal distribution with evenly spread scores.
   - **Reasoning score -** Normal distribution of scores.
2. **Boxplots:-**
   - **Salary -** Many extroverts at the highest end, suggest some candidates with exceptionally high salaries.
   - **College CGPA, English scores, meaningful scores -** No outliers found, with median values near the middle of the interquartile range.
3. **Bar Plot:-**
   - **Gender -** A balanced distribution between male and female candidates.
   - **College Tier -** More candidates come from Tier 2 colleges as compared to Tier 1 colleges.
   - **Degrees -** Most candidates have graduated.

5 4. BIVARIATE ANALYSIS

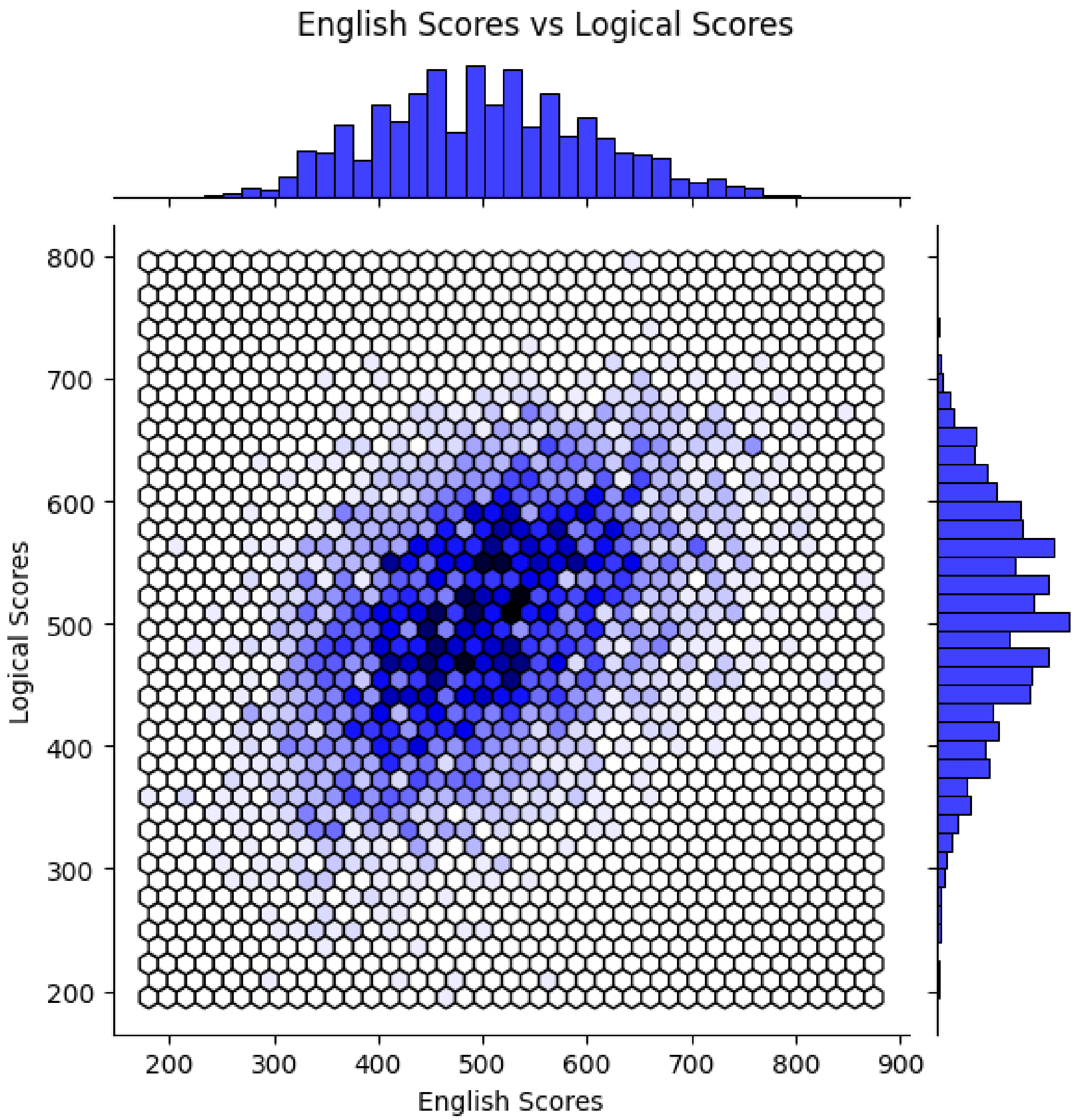
[73]:
```
# Relationships between Numerical Columns

pluto.figure(figsize=(10, 6))
sobu.scatterplot(x='collegeGPA', y='Salary', data=dataframe, color='red',␣
↪edgecolor='black')
pluto.title('College CGPA vs Salary')
pluto.xlabel('College CGPA')
pluto.ylabel('Salary')
```

```
pluto.show() # Scatter Plot
```

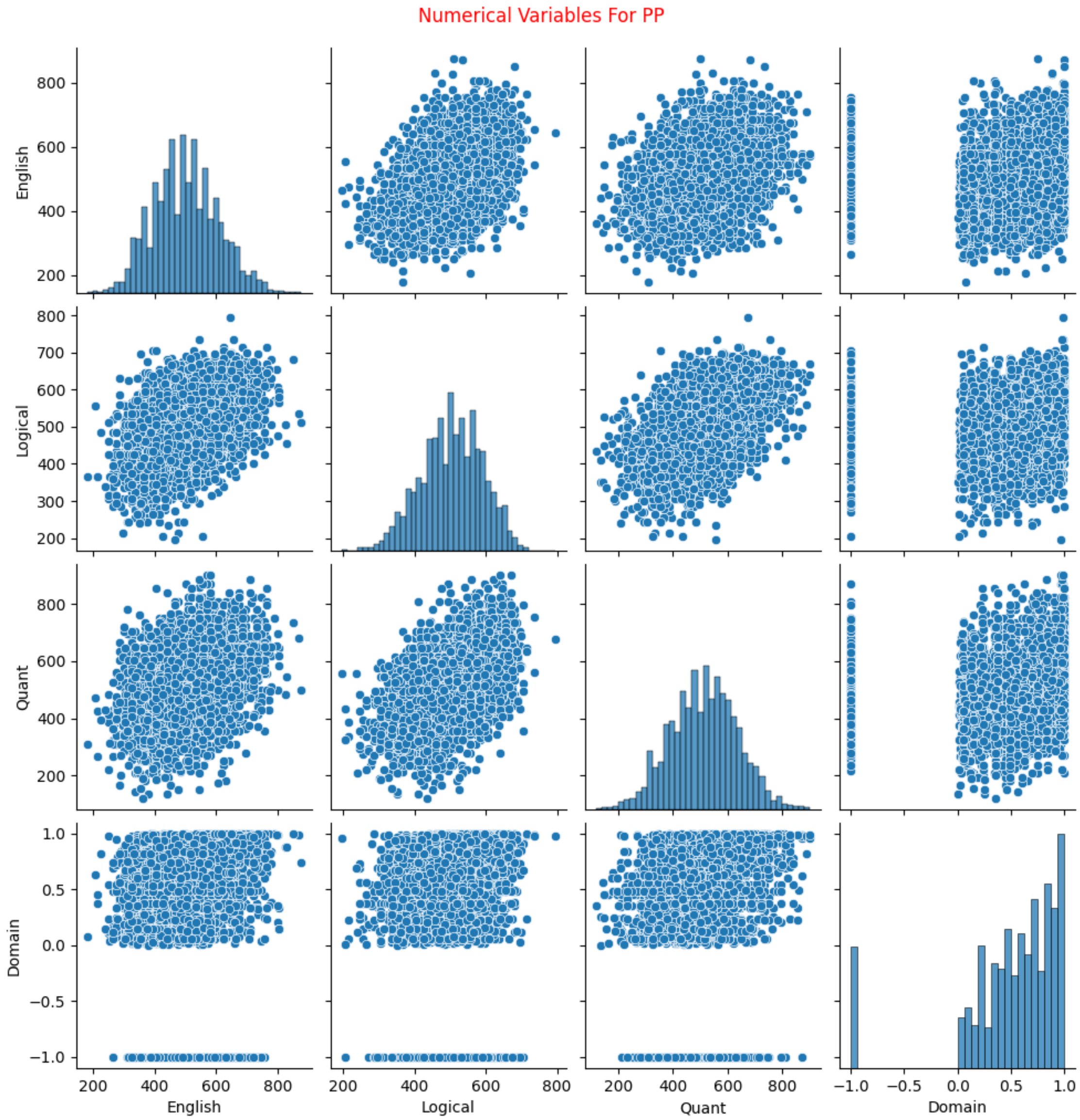


[65]:
```
pluto.figure(figsize=(10, 6))
sobu.jointplot(x='English', y='Logical', data=dataframe, kind='hex',␣
 ↪color='blue', edgecolor='black')
pluto.suptitle('English Scores vs Logical Scores', y=1.02)
pluto.xlabel('English Scores')
pluto.ylabel('Logical Scores')
pluto.show() # Hexbin Plot
```

```
<Figure size 1000x600 with 0 Axes>
```



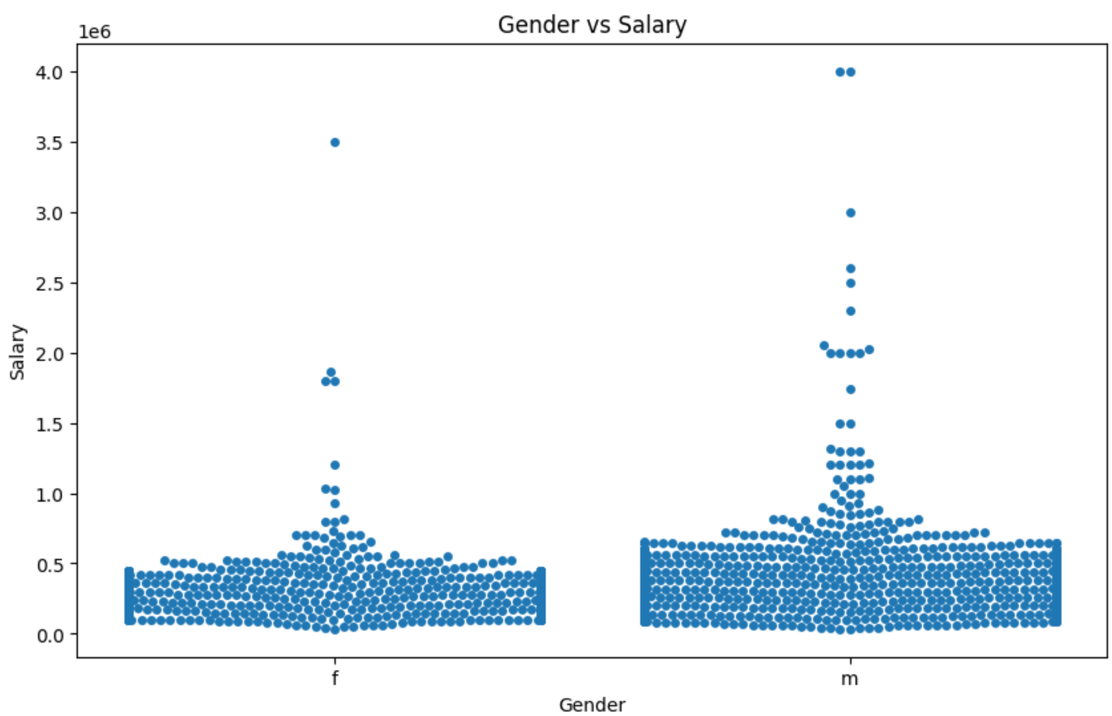
[70]:
```
sobu.pairplot(dataframe[['English', 'Logical', 'Quant', 'Domain']])
pluto.suptitle('Numerical Variables For PP', y=1.02, color='red')
pluto.show()
```



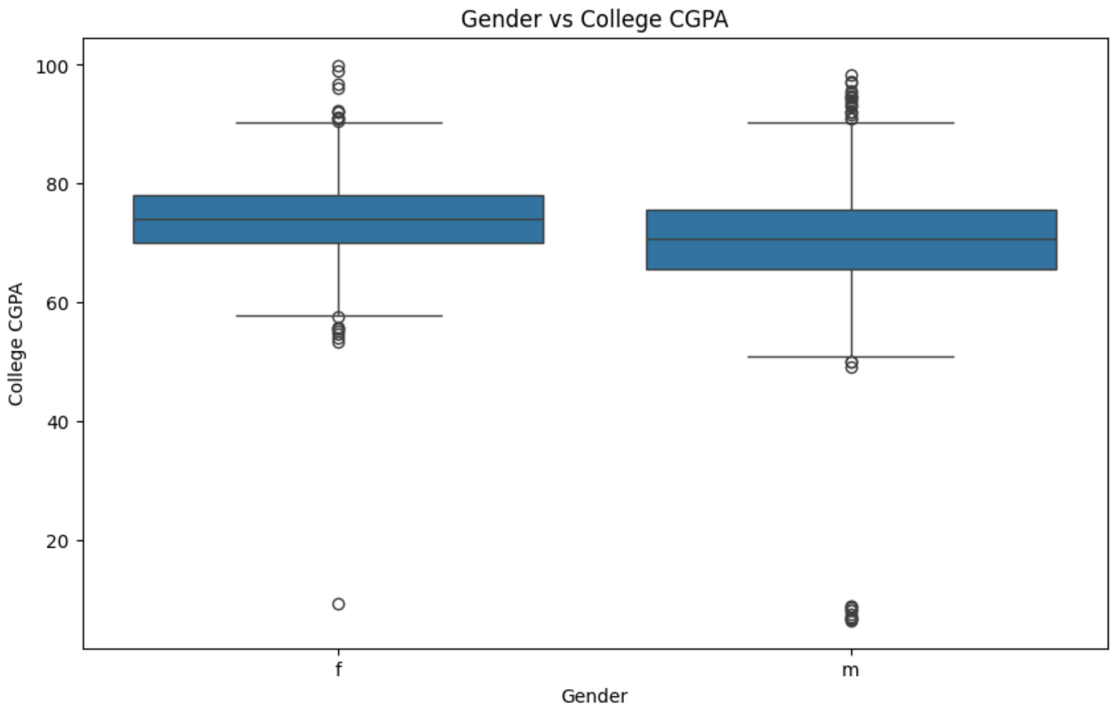
[77]:
```
# Relationships between Categorical and Numerical Columns

pluto.figure(figsize=(10, 6))
sobu.swarmplot(x='Gender', y='Salary', data=dataframe)
pluto.title('Gender vs Salary')
pluto.xlabel('Gender')
pluto.ylabel('Salary')
pluto.show()  # Swarm Plot
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 41.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
```

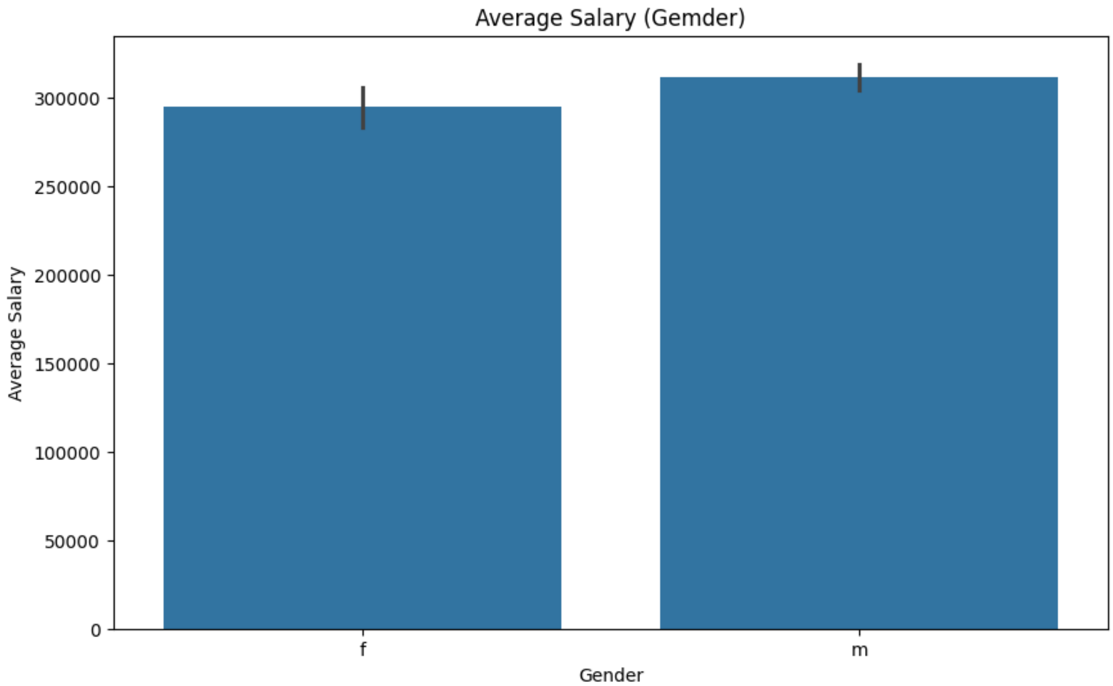
```
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 72.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 62.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 82.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



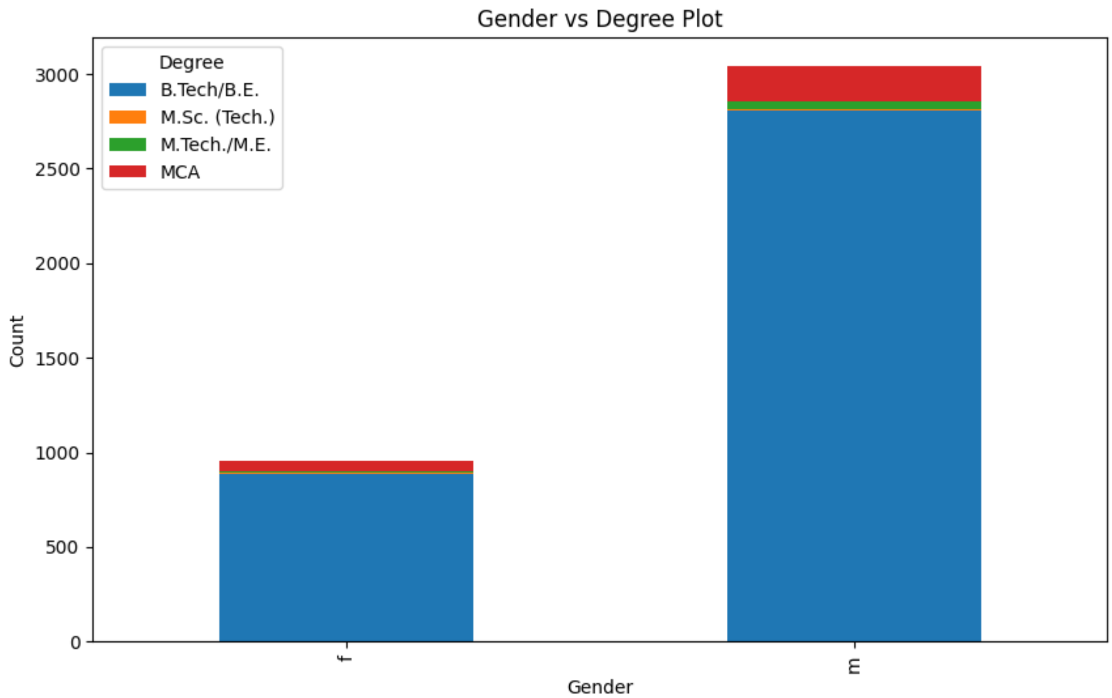
[76]:
```python
pluto.figure(figsize=(10, 6))
sobu.boxplot(x='Gender', y='collegeGPA', data=dataframe)
pluto.title('Gender vs College CGPA')
pluto.xlabel('Gender')
pluto.ylabel('College CGPA')
pluto.show()  # Box Plot
```



[78]:
```
pluto.figure(figsize=(10, 6))
sobu.barplot(x='Gender', y='Salary', data=dataframe)
pluto.title('Average Salary (Gemder)')
pluto.xlabel('Gender')
pluto.ylabel('Average Salary')
pluto.show() # Bar Plot
```



[82]:
```
# Relationships between Categorical Columns

cross_tab = pandu.crosstab(dataframe['Gender'], dataframe['Degree'])
cross_tab.plot(kind='bar', stacked=True, figsize=(10, 6))
pluto.title('Gender vs Degree Plot')
pluto.xlabel('Gender')
pluto.ylabel('Count')
pluto.show()  # Stacked Bar Plot
```



1. **Scatter plot of college GPA and salary:-**
   - Found a positive correlation between college CGPA and salary.
2. **Hexbin plot of English scores vs. English scores. comprehension score:-**
   - Positive correlation between English scores and comprehension scores.
3. **Pair plot of statistical variables:-**
   - Identify distributional and pairwise relationships between statistical variables.
4. **Swarm plot of gender and salary:=**
   - More data points for men than women.
   - The distribution of wages for men is relatively wide.
   - Some outliers from both genders.
5. **Box Plot of Gender and College CGPA:-**
   - Slightly higher median college CGPA for women.
   - Similar changes in college CGPA between men and women.
6. **Bar plot of average salary by gender:-**
   - Men get paid slightly more.
   - Some combine between faiths.
7. **Stacked bar plot of gender and degree:**
   - Many men with bachelor's degrees.
   - More women with master's degrees.

6 5. RESEARCH QUESTION

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

[89]:
```
specificjobs = ['senior quality engineer']
csescandidates = dataframe[(dataframe['Degree'] == 'B.Tech/B.E.') &␣
 ↪(dataframe['Specialization'] == 'computer engineering')]
filtereddf = csescandidates[csescandidates['Designation'].isin(specificjobs)]
avgsalaries = filtereddf['Salary'].mean()
print("Average Salary for CSE Freshers in Specific Role:-", avgsalaries, "LPA")
```

```
Average Salary for CSE Freshers in Specific Role:- 460000.0 LPA
```

7 6. CONCLUSION

the analysis provided insights into the dataset containing employment outcomes of engineering graduates, focusing on salary, gender, specialization, and other relevant variables. Here are the key findings:-

1. Salary Analysis:- The average salary for CSE Freshers in specific roles (e.g., senior quality engineer) was found to be 4,60,000 LPA.
2. Gender and Specialization:- There was a distribution of specialization by gender, indicating potential differences in specialization preferences between males and females. However, further analysis is required to determine the significance of this relationship.

Overall, the analysis sheds light on the employment outcomes and characteristics of engineering graduates, providing valuable insights for further research and decision-making processes in the field. Further exploration and analysis could enhance understanding and provide more comprehensive insights into the factors influencing employment outcomes for engineering graduates.

8 7. (BONUS) COME UP WITH SOME INTERESTING CONCLUSION & RESEARCH QUESTION

First, the average salary for computer and other engineering graduates in specific roles such as Senior Quality Engineer is 460,000 LPA This finding highlights the income of new graduates in engineering can get emphasized and is a criterion for wage expectations.

Furthermore, the study revealed interesting patterns of gender preference. Although further research is needed, preliminary findings suggest that there may be differences in key options for men and women. Understanding these differences can inform targeted initiatives to promote diversity and inclusion in the technology workforce.

Moving forward, several interesting research questions emerge from the study. Examining the effects of specialization on wages, local differences in employment opportunities, and the effects of college status on employment trajectories can provide deeper insights into the factors affecting employment outcomes that they of the completed technology acquisition. Additionally, gender research differences in wage negotiations, career development trajectories over time, and the effects of personality traits on career success may shed light on important areas a technological advances must be made.

Overall, this study serves as a starting point for further research and analysis to understand the dynamics of engineering graduate employment outcomes. By addressing these research questions, stakeholders can make informed decisions to enhance career prospects and promote equal opportunities in technology.