amcat-eda

October 4, 2024

1 DATA ANALYSIS

[26]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2 LOAD THE DATA

[27]:
```
data   = pd.read_excel("D:\INNOMATICS FILE\data.xlsx")
```

[28]:
```
data.head()
```

[28]:

	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	…

	Designation	JobCity	Gender	DOB	10percentage	…
0	senior quality engineer	Bangalore	f	1990-02-19	84.3	…
1	assistant manager	Indore	m	1989-10-04	85.4	…
2	systems engineer	Chennai	f	1992-08-03	85.0	…
3	senior software engineer	Gurgaon	m	1989-12-05	85.6	…
4	get	Manesar	m	1991-02-27	78.0	…

	ComputerScience	MechanicalEngg	ElectricalEngg	TelecomEngg	CivilEngg
0	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	-1	-1	-1	-1	-1

	conscientiousness	agreeableness	extraversion	nueroticism
0	0.9737	0.8128	0.5269	1.35490
1	-0.7335	0.3789	1.2396	-0.10760

```
2             0.2718        1.7109        0.1637     -0.86820
3             0.0464        0.3448       -0.3440     -0.40780
4            -0.8810       -0.2793       -1.0697      0.09163

   openess_to_experience
0                -0.4455
1                 0.8637
2                 0.6721
3                -0.9194
4                -0.1295

[5 rows x 39 columns]
```

3 Basic data overview: head, shape, and description

[53]:
```
overview = {
    "Head": data.head(),
    "Shape": data.shape,
    "Description": data.describe(include='all')
}

overview
```

```
C:\Users\ranji\AppData\Local\Temp\ipykernel_19984\683686885.py:5: FutureWarning:
Treating datetime data as categorical rather than numeric in `.describe` is
deprecated and will be removed in a future version of pandas. Specify
`datetime_is_numeric=True` to silence this warning and adopt the future behavior
now.
  "Description": data.describe(include='all')
C:\Users\ranji\AppData\Local\Temp\ipykernel_19984\683686885.py:5: FutureWarning:
Treating datetime data as categorical rather than numeric in `.describe` is
deprecated and will be removed in a future version of pandas. Specify
`datetime_is_numeric=True` to silence this warning and adopt the future behavior
now.
  "Description": data.describe(include='all')
```

[53]:
```
{'Head':   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

                 Designation    JobCity Gender        DOB  10percentage  …  \
 0   senior quality engineer  Bangalore      f 1990-02-19          84.3  …
 1         assistant manager     Indore      m 1989-10-04          85.4  …
```

```
 2          systems engineer    Chennai      f 1992-08-03          85.0  …
 3  senior software engineer    Gurgaon      m 1989-12-05          85.6  …
 4                       get    Manesar      m 1991-02-27          78.0  …

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

    conscientiousness agreeableness extraversion  nueroticism  \
 0             0.9737        0.8128       0.5269      1.35490
 1            -0.7335        0.3789       1.2396     -0.10760
 2             0.2718        1.7109       0.1637     -0.86820
 3             0.0464        0.3448      -0.3440     -0.40780
 4            -0.8810       -0.2793      -1.0697      0.09163

    openess_to_experience
 0                -0.4455
 1                 0.8637
 2                 0.6721
 3                -0.9194
 4                -0.1295

 [5 rows x 39 columns],
 'Shape': (3998, 39),
 'Description':        Unnamed: 0            ID        Salary
DOJ      DOL  \
 count        3998  3.998000e+03  3.998000e+03                 3998     3998
 unique          1           NaN           NaN                   81       67
 top         train           NaN           NaN  2014-07-01 00:00:00  present
 freq         3998           NaN           NaN                  199     1875
 first         NaN           NaN           NaN  1991-06-01 00:00:00      NaN
 last          NaN           NaN           NaN  2015-12-01 00:00:00      NaN
 mean          NaN  6.637945e+05  3.076998e+05                  NaN      NaN
 std           NaN  3.632182e+05  2.127375e+05                  NaN      NaN
 min           NaN  1.124400e+04  3.500000e+04                  NaN      NaN
 25%           NaN  3.342842e+05  1.800000e+05                  NaN      NaN
 50%           NaN  6.396000e+05  3.000000e+05                  NaN      NaN
 75%           NaN  9.904800e+05  3.700000e+05                  NaN      NaN
 max           NaN  1.298275e+06  4.000000e+06                  NaN      NaN

              Designation    JobCity Gender                  DOB  \
 count               3998       3998   3998                 3998
 unique               419        339      2                 1872
 top     software engineer  Bangalore      m  1991-01-01 00:00:00
```

freq	539	627	3041		11
first	NaN	NaN	NaN	1977-10-30	00:00:00
last	NaN	NaN	NaN	1997-05-27	00:00:00
mean	NaN	NaN	NaN		NaN
std	NaN	NaN	NaN		NaN
min	NaN	NaN	NaN		NaN
25%	NaN	NaN	NaN		NaN
50%	NaN	NaN	NaN		NaN
75%	NaN	NaN	NaN		NaN
max	NaN	NaN	NaN		NaN

	10percentage	…	ComputerScience	MechanicalEngg	ElectricalEngg	\
count	3998.000000	…	3998.000000	3998.000000	3998.000000	
unique	NaN	…	NaN	NaN	NaN	
top	NaN	…	NaN	NaN	NaN	
freq	NaN	…	NaN	NaN	NaN	
first	NaN	…	NaN	NaN	NaN	
last	NaN	…	NaN	NaN	NaN	
mean	77.925443	…	90.742371	22.974737	16.478739	
std	9.850162	…	175.273083	98.123311	87.585634	
min	43.000000	…	-1.000000	-1.000000	-1.000000	
25%	71.680000	…	-1.000000	-1.000000	-1.000000	
50%	79.150000	…	-1.000000	-1.000000	-1.000000	
75%	85.670000	…	-1.000000	-1.000000	-1.000000	
max	97.760000	…	715.000000	623.000000	676.000000	

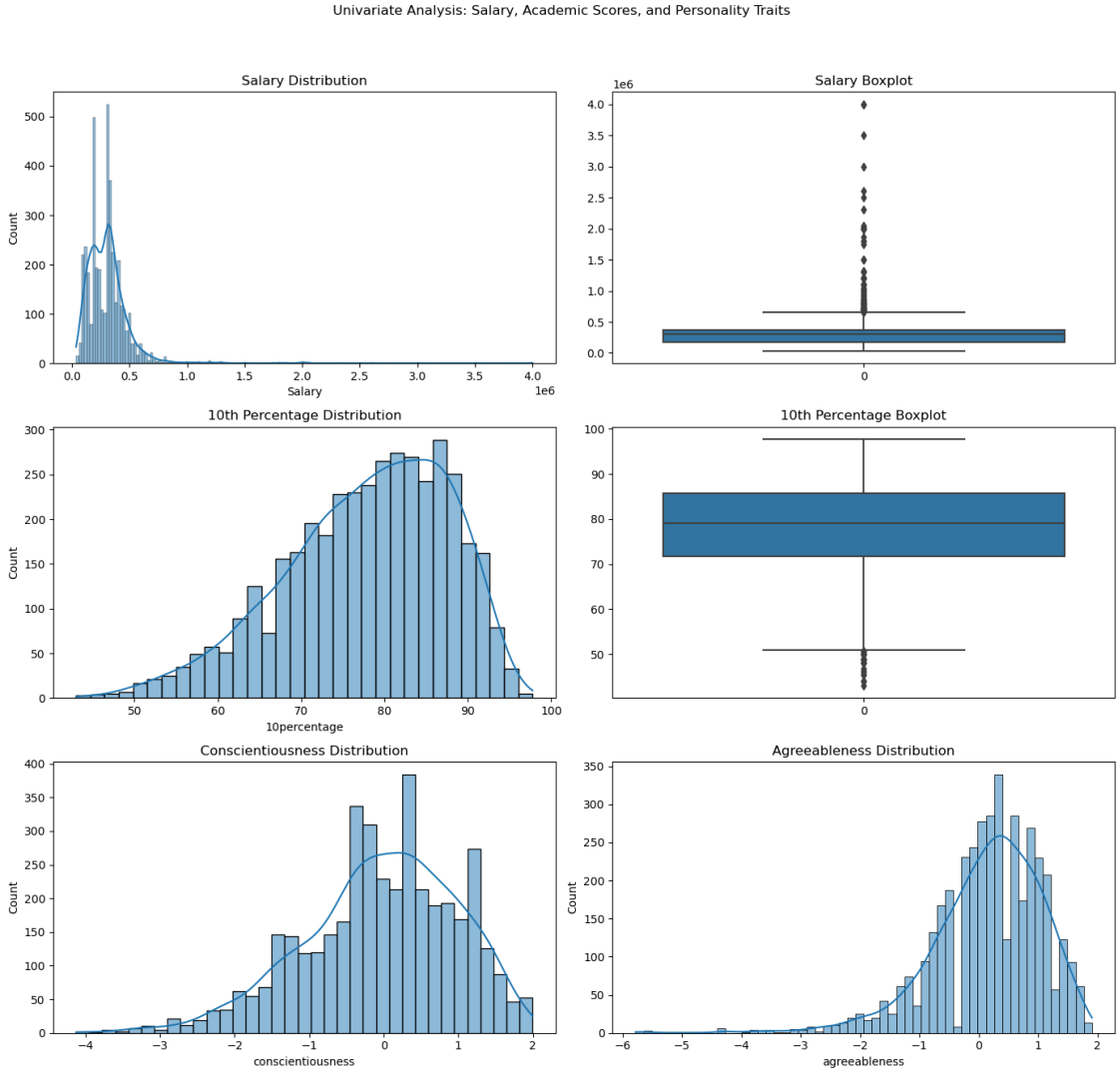
	TelecomEngg	CivilEngg	conscientiousness	agreeableness	\
count	3998.000000	3998.000000	3998.000000	3998.000000	
unique	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	
first	NaN	NaN	NaN	NaN	
last	NaN	NaN	NaN	NaN	
mean	31.851176	2.683842	-0.037831	0.146496	
std	104.852845	36.658505	1.028666	0.941782	
min	-1.000000	-1.000000	-4.126700	-5.781600	
25%	-1.000000	-1.000000	-0.713525	-0.287100	
50%	-1.000000	-1.000000	0.046400	0.212400	
75%	-1.000000	-1.000000	0.702700	0.812800	
max	548.000000	516.000000	1.995300	1.904800	

	extraversion	nueroticism	openess_to_experience
count	3998.000000	3998.000000	3998.000000
unique	NaN	NaN	NaN
top	NaN	NaN	NaN
freq	NaN	NaN	NaN
first	NaN	NaN	NaN

```
last            NaN          NaN                  NaN
mean       0.002763    -0.169033            -0.138110
std        0.951471     1.007580             1.008075
min       -4.600900    -2.643000            -7.375700
25%       -0.604800    -0.868200            -0.669200
50%        0.091400    -0.234400            -0.094300
75%        0.672000     0.526200             0.502400
max        2.535400     3.352500             1.822400

[13 rows x 39 columns]}
```

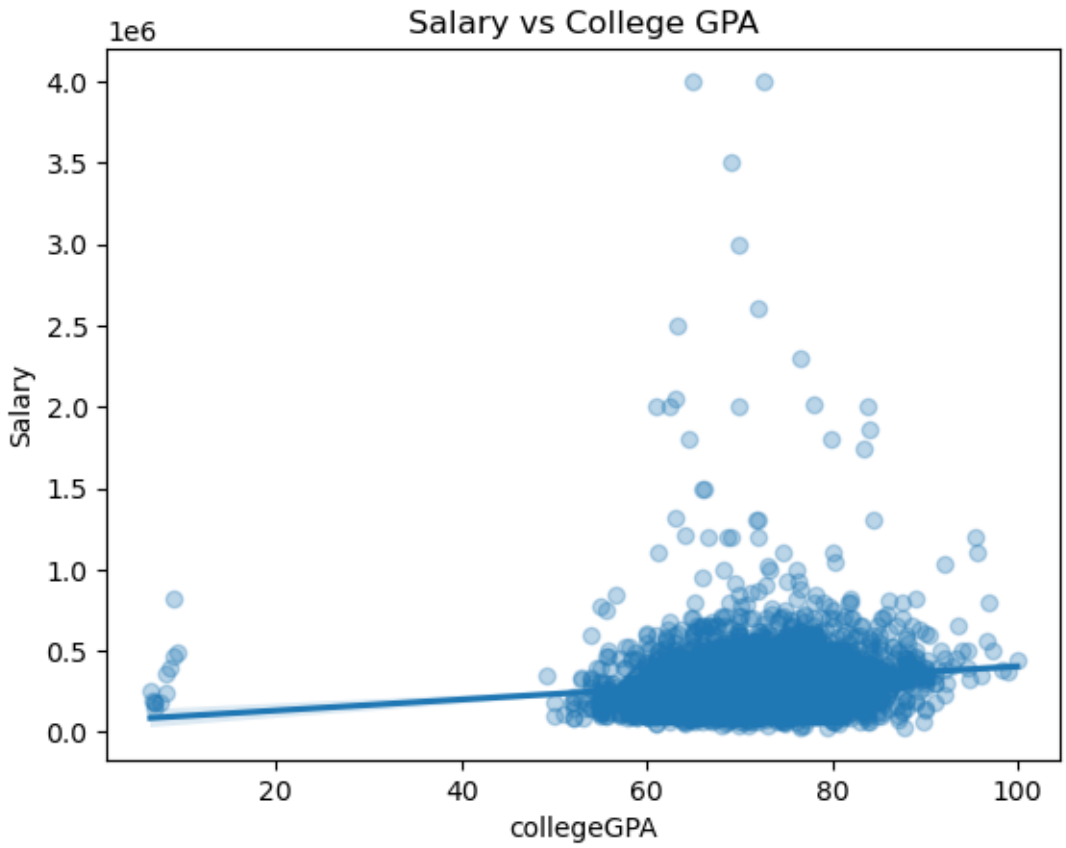
4 Univariate Analysis:

[30]:
```
# 1. Salary Distribution Analysis
fig, axes = plt.subplots(3, 2, figsize=(14, 14))
fig.suptitle("Univariate Analysis: Salary, Academic Scores, and Personality␣
 ↪Traits")

# Salary - Histogram and Boxplot
sns.histplot(data['Salary'], kde=True, ax=axes[0, 0])
axes[0, 0].set_title("Salary Distribution")

sns.boxplot(data['Salary'], ax=axes[0, 1])
axes[0, 1].set_title("Salary Boxplot")

# 10th Percentage - Histogram and Boxplot
sns.histplot(data['10percentage'], kde=True, ax=axes[1, 0])
axes[1, 0].set_title("10th Percentage Distribution")

sns.boxplot(data['10percentage'], ax=axes[1, 1])
axes[1, 1].set_title("10th Percentage Boxplot")

# Personality Trait: Conscientiousness - Histogram
sns.histplot(data['conscientiousness'], kde=True, ax=axes[2, 0])
axes[2, 0].set_title("Conscientiousness Distribution")

# Personality Trait: Agreeableness - Histogram
sns.histplot(data['agreeableness'], kde=True, ax=axes[2, 1])
axes[2, 1].set_title("Agreeableness Distribution")

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



5 Bivariate Analysis

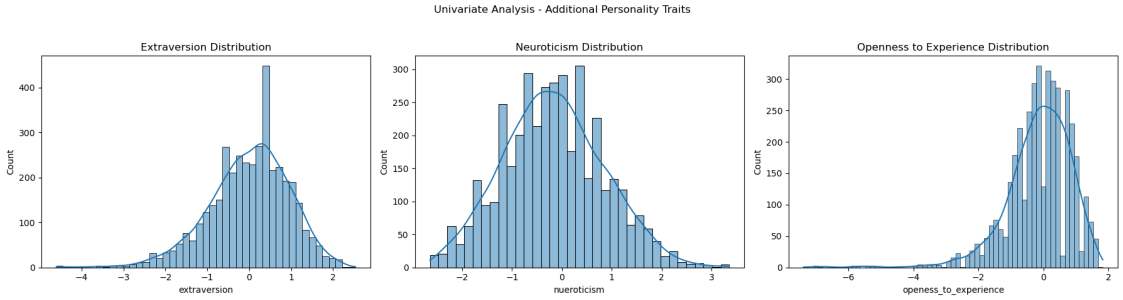
[40]:
```
sns.regplot(x='collegeGPA', y='Salary', data=data, scatter_kws={'alpha':0.3})
plt.title("Salary vs College GPA")
plt.show()
```



6 Additional Personality Traits:

[32]:
```
# Set up figure for additional personality traits
fig, axes = plt.subplots(1, 3, figsize=(18, 5))
fig.suptitle("Univariate Analysis - Additional Personality Traits")

# Extraversion - Histogram
sns.histplot(data['extraversion'], kde=True, ax=axes[0])
axes[0].set_title("Extraversion Distribution")

# Neuroticism - Histogram
sns.histplot(data['nueroticism'], kde=True, ax=axes[1])
axes[1].set_title("Neuroticism Distribution")

# Openness to Experience - Histogram
sns.histplot(data['openess_to_experience'], kde=True, ax=axes[2])
axes[2].set_title("Openness to Experience Distribution")

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

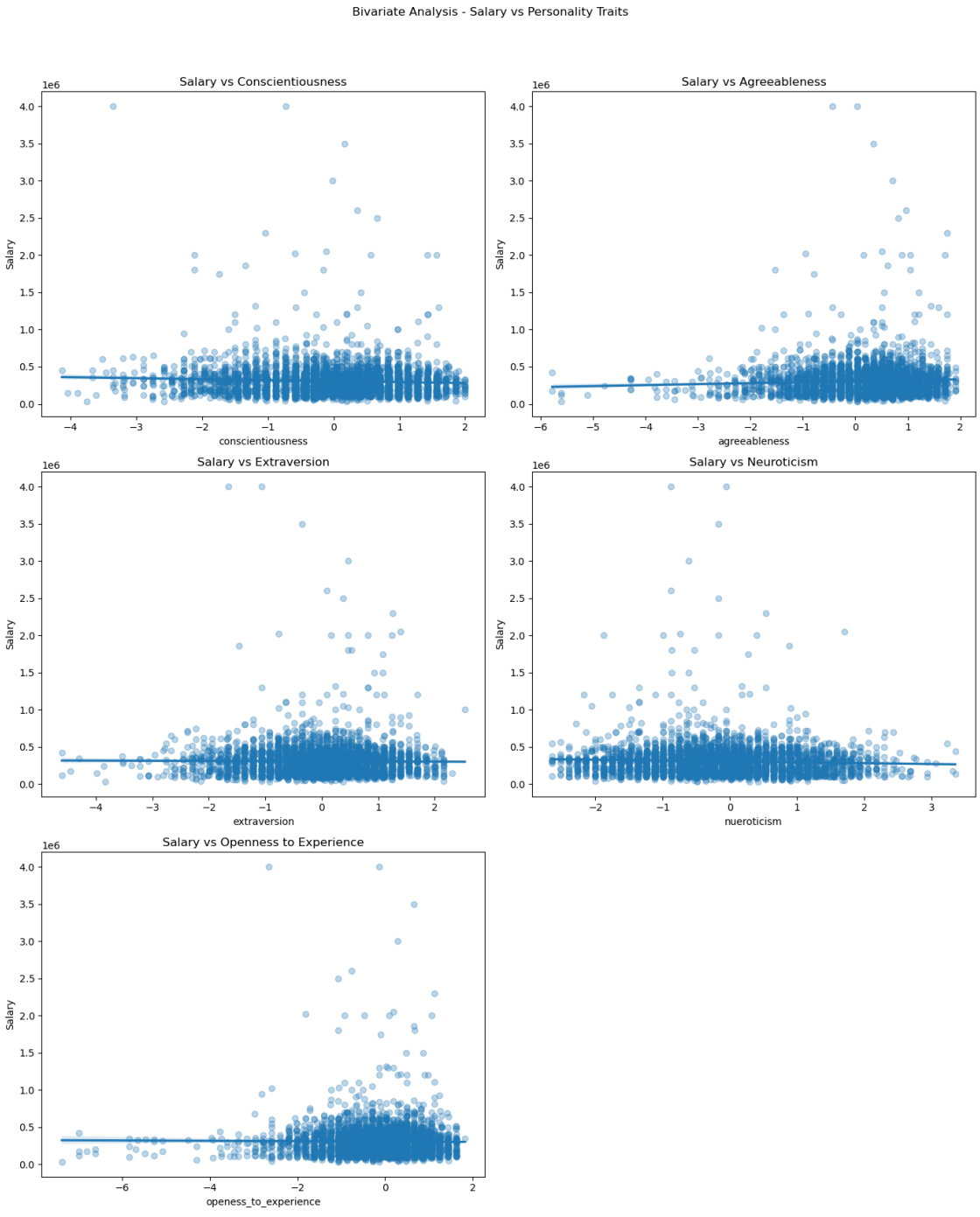
```
plt.show()
```



7 Relationships between Salary and Personality Traits:

[33]:
```
# relationships between Salary and each personality trait
fig, axes = plt.subplots(3, 2, figsize=(14, 18))
fig.suptitle("Bivariate Analysis - Salary vs Personality Traits")

# Salary vs Conscientiousness
sns.regplot(x='conscientiousness', y='Salary', data=data, ax=axes[0, 0],␣
 ↪scatter_kws={'alpha':0.3})
axes[0, 0].set_title("Salary vs Conscientiousness")

# Salary vs Agreeableness
sns.regplot(x='agreeableness', y='Salary', data=data, ax=axes[0, 1],␣
 ↪scatter_kws={'alpha':0.3})
axes[0, 1].set_title("Salary vs Agreeableness")

# Salary vs Extraversion
sns.regplot(x='extraversion', y='Salary', data=data, ax=axes[1, 0],␣
 ↪scatter_kws={'alpha':0.3})
axes[1, 0].set_title("Salary vs Extraversion")

# Salary vs Neuroticism
sns.regplot(x='nueroticism', y='Salary', data=data, ax=axes[1, 1],␣
 ↪scatter_kws={'alpha':0.3})
axes[1, 1].set_title("Salary vs Neuroticism")

# Salary vs Openness to Experience
sns.regplot(x='openess_to_experience', y='Salary', data=data, ax=axes[2, 0],␣
 ↪scatter_kws={'alpha':0.3})
axes[2, 0].set_title("Salary vs Openness to Experience")

# Turn off unused subplot
```

```
axes[2, 1].axis('off')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

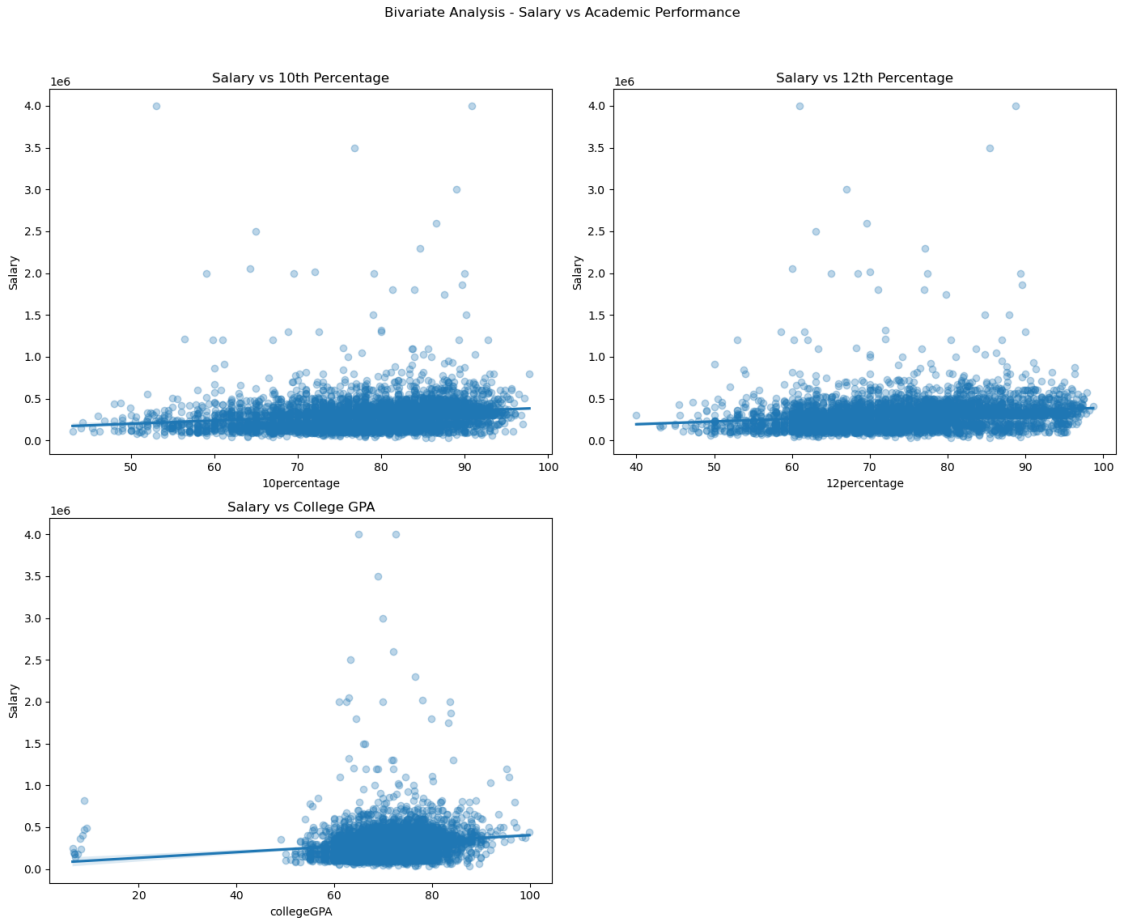


[35]:
```
#  identify academic performance-related columns
data.columns
```

[35]:
```
Index(['Unnamed: 0', 'ID', 'Salary', 'DOJ', 'DOL', 'Designation', 'JobCity',
       'Gender', 'DOB', '10percentage', '10board', '12graduation',
       '12percentage', '12board', 'CollegeID', 'CollegeTier', 'Degree',
       'Specialization', 'collegeGPA', 'CollegeCityID', 'CollegeCityTier',
       'CollegeState', 'GraduationYear', 'English', 'Logical', 'Quant',
       'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
       'CivilEngg', 'conscientiousness', 'agreeableness', 'extraversion',
       'nueroticism', 'openess_to_experience'],
      dtype='object')
```

8 Relationships between Salary and Academic Performance:

[36]:
```
#  relationships between Salary and each academic performance metric
fig, axes = plt.subplots(2, 2, figsize=(14, 12))
fig.suptitle("Bivariate Analysis - Salary vs Academic Performance")

# Salary vs 10th Percentage
sns.regplot(x='10percentage', y='Salary', data=data, ax=axes[0, 0],␣
 ↪scatter_kws={'alpha':0.3})
axes[0, 0].set_title("Salary vs 10th Percentage")

# Salary vs 12th Percentage
sns.regplot(x='12percentage', y='Salary', data=data, ax=axes[0, 1],␣
 ↪scatter_kws={'alpha':0.3})
axes[0, 1].set_title("Salary vs 12th Percentage")

# Salary vs College GPA
sns.regplot(x='collegeGPA', y='Salary', data=data, ax=axes[1, 0],␣
 ↪scatter_kws={'alpha':0.3})
axes[1, 0].set_title("Salary vs College GPA")

# Turn off unused subplot
axes[1, 1].axis('off')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



9 Impact of College GPA on Salary

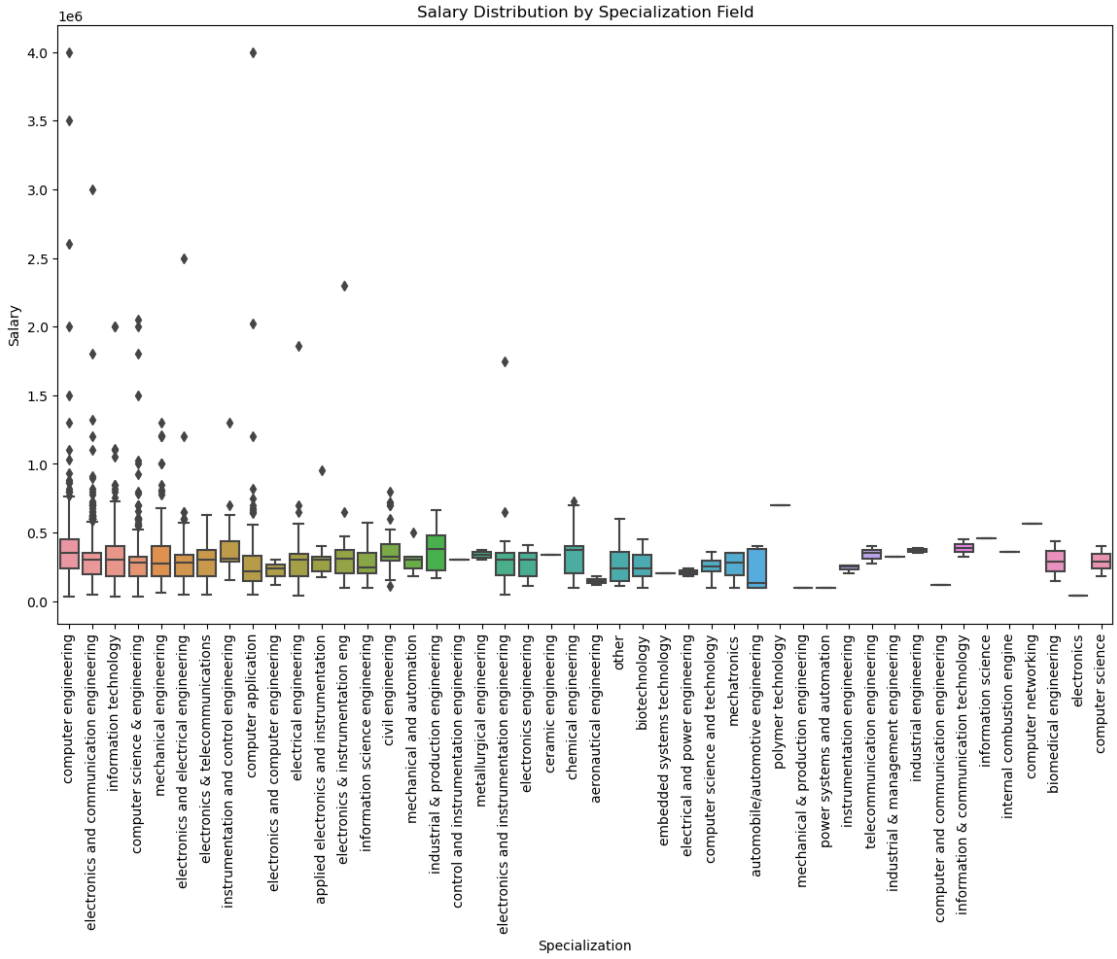
[37]:
```
import statsmodels.api as sm

#  linear regression analysis to quantify the impact of College GPA on Salary
X = sm.add_constant(data['collegeGPA'])
y = data['Salary']

# Fit the regression model
model = sm.OLS(y, X).fit()
regression_summary = model.summary()

regression_summary
```

[37]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.017
Model:                            OLS   Adj. R-squared:                  0.017
Method:                 Least Squares   F-statistic:                     68.80
Date:                Fri, 04 Oct 2024   Prob (F-statistic):           1.47e-16
Time:                        22:02:16   Log-Likelihood:                -54685.
No. Observations:                3998   AIC:                         1.094e+05
Df Residuals:                    3996   BIC:                         1.094e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       6.545e+04   2.94e+04      2.226      0.026    7814.044    1.23e+05
collegeGPA  3388.8258    408.549      8.295      0.000    2587.843    4189.809
==============================================================================
Omnibus:                     5019.550   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1216344.373
Skew:                           6.638   Prob(JB):                         0.00
Kurtosis:                      87.412   Cond. No.                         634.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

10 Salary Distribution by Specialization Field:

[38]:
```
# Set up a figure for the boxplot
plt.figure(figsize=(14, 8))
sns.boxplot(x='Specialization', y='Salary', data=data)
plt.title("Salary Distribution by Specialization Field")
plt.xticks(rotation=90)
plt.show()
```



11 Salary Distribution by Graduation Year:

[39]:
```
# Set up a figure for the boxplot
plt.figure(figsize=(14, 8))
sns.boxplot(x='GraduationYear', y='Salary', data=data)
plt.title("Salary Distribution by Graduation Year")
plt.xticks(rotation=45)
plt.show()
```

12 Salary Claim Validation:

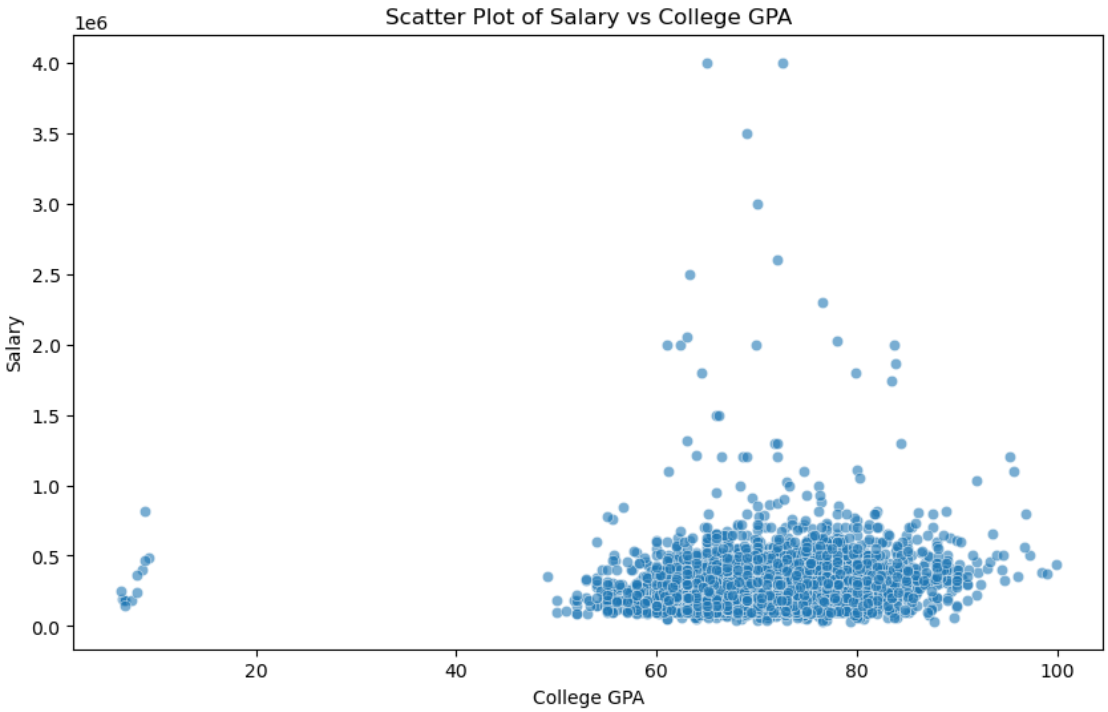
[42]:
```
cs_engineering = data[(data['Specialization'] == 'Computer Science') &
                      (data['Designation'].isin(['Programming Analyst',␣
 ↪'Software Engineer', 'Hardware Engineer', 'Associate Engineer']))]
cs_engineering['Salary'].describe()
```

[42]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: Salary, dtype: float64
```

13 Scatter Plot of Salary vs. College GPA

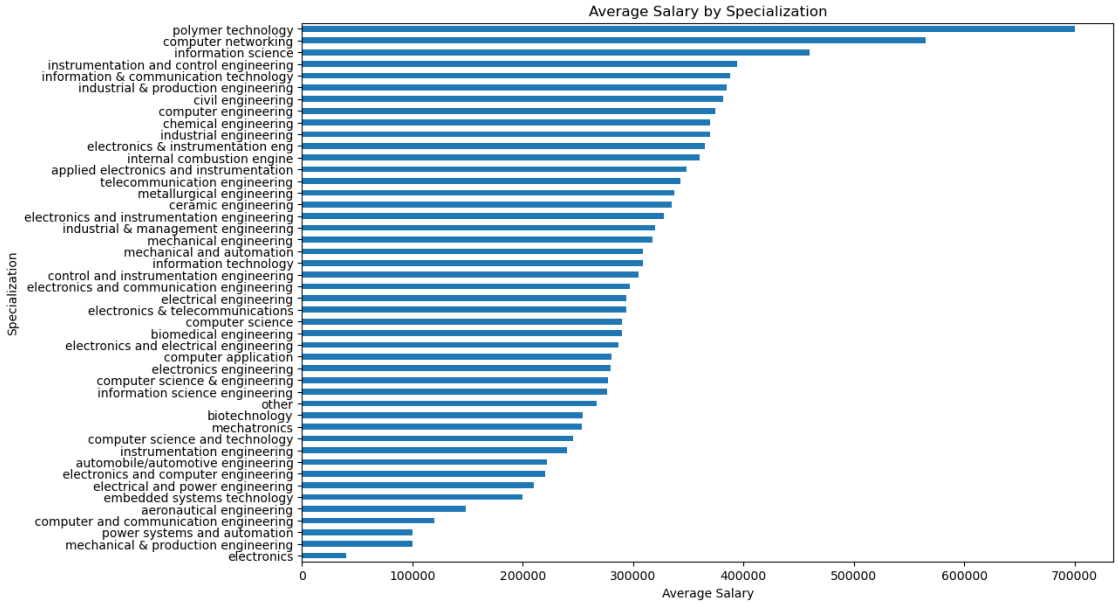
[43]:
```
# Scatter plot for Salary vs College GPA
plt.figure(figsize=(10, 6))
sns.scatterplot(x='collegeGPA', y='Salary', data=data, alpha=0.6)
```

```
plt.title("Scatter Plot of Salary vs College GPA")
plt.xlabel("College GPA")
plt.ylabel("Salary")
plt.show()
```



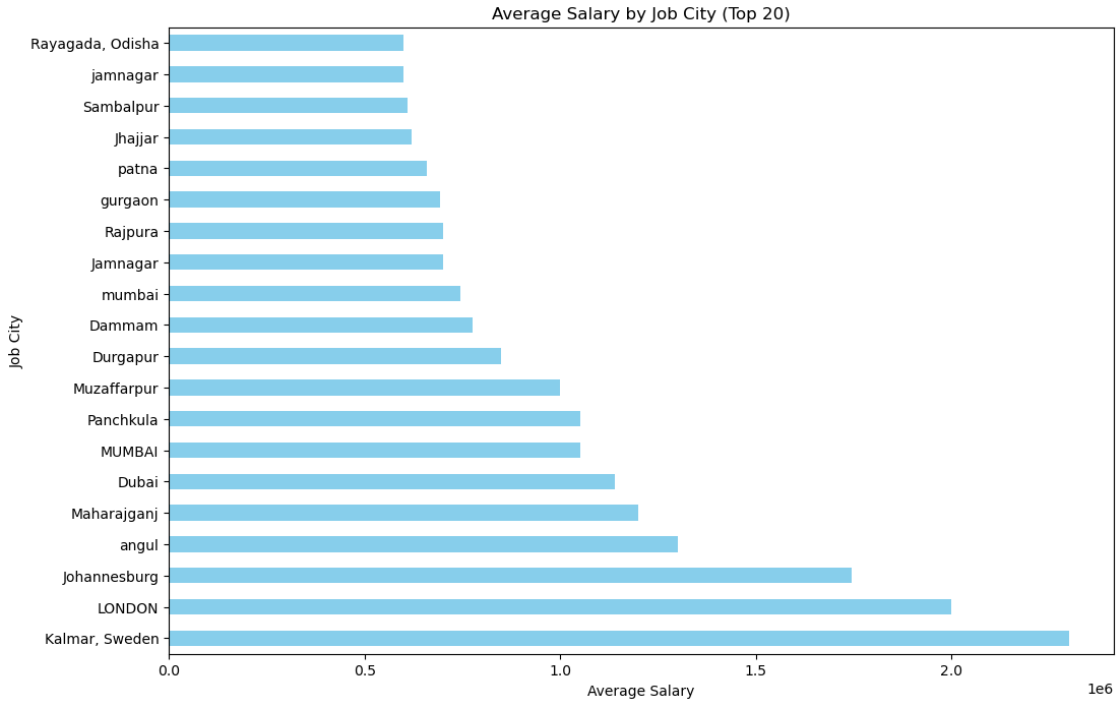
14 Bar Chart of Average Salary by Specialization

[44]:
```
# Calculate the mean salary for each specialization
avg_salary_by_specialization = data.groupby('Specialization')['Salary'].mean().
 ↪sort_values()

# Plot the average salary by specialization
plt.figure(figsize=(12, 8))
avg_salary_by_specialization.plot(kind='barh')
plt.title("Average Salary by Specialization")
plt.xlabel("Average Salary")
plt.ylabel("Specialization")
plt.show()
```



15 Bar chart of average salary by Job City

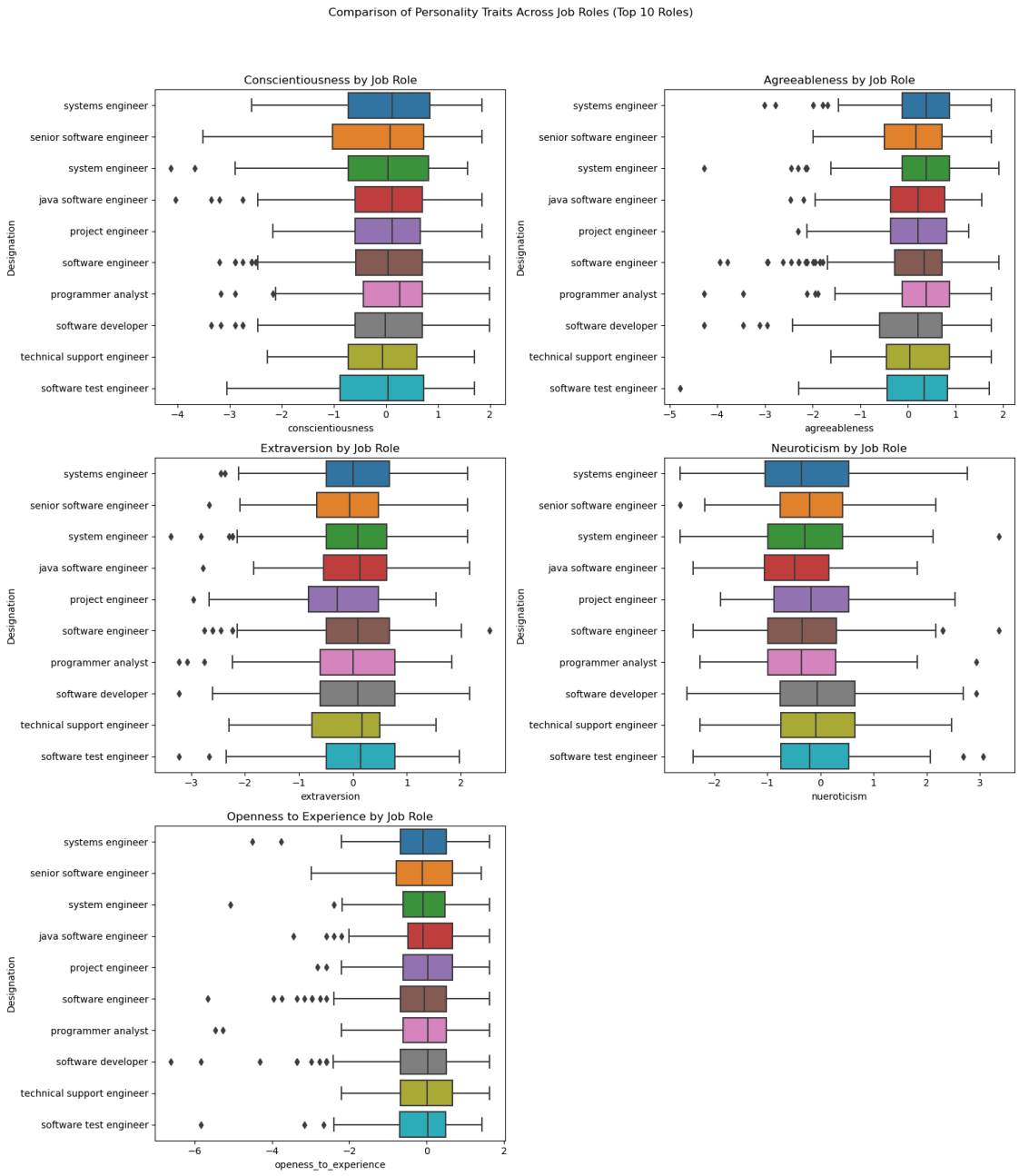
[45]:
```
avg_salary_by_city = data.groupby('JobCity')['Salary'].mean().
↪sort_values(ascending=False).head(20)

# Plot the average salary by job city
plt.figure(figsize=(12, 8))
avg_salary_by_city.plot(kind='barh', color="skyblue")
plt.title("Average Salary by Job City (Top 20)")
plt.xlabel("Average Salary")
plt.ylabel("Job City")
plt.show()
```



16 Personality Traits Across Job Roles (Top 10 Roles)

[47]:
```
top_roles = data['Designation'].value_counts().head(10).index
role_data = data[data['Designation'].isin(top_roles)]

# Set up the figure for personality traits across job roles
fig, axes = plt.subplots(3, 2, figsize=(15, 18))
fig.suptitle("Comparison of Personality Traits Across Job Roles (Top 10 Roles)")

# Conscientiousness by Job Role
sns.boxplot(x='conscientiousness', y='Designation', data=role_data, ax=axes[0,␣
↪0])
axes[0, 0].set_title("Conscientiousness by Job Role")

# Agreeableness by Job Role
sns.boxplot(x='agreeableness', y='Designation', data=role_data, ax=axes[0, 1])
axes[0, 1].set_title("Agreeableness by Job Role")

# Extraversion by Job Role
sns.boxplot(x='extraversion', y='Designation', data=role_data, ax=axes[1, 0])
axes[1, 0].set_title("Extraversion by Job Role")

# Neuroticism by Job Role
```

```
sns.boxplot(x='nueroticism', y='Designation', data=role_data, ax=axes[1, 1])
axes[1, 1].set_title("Neuroticism by Job Role")

# Openness to Experience by Job Role
sns.boxplot(x='openess_to_experience', y='Designation', data=role_data,␣
 ↪ax=axes[2, 0])
axes[2, 0].set_title("Openness to Experience by Job Role")

# Turn off the last unused subplot
axes[2, 1].axis('off')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



17 Correlation Between College GPA and Personality Traits

[48]:
```
#Selecting the relevant columns
gpa_trait_data = data[['collegeGPA', 'conscientiousness', 'agreeableness',␣
 ↪'extraversion', 'nueroticism', 'openess_to_experience']]

# Compute correlation matrix
```

```
correlation_matrix = gpa_trait_data.corr()

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation between College GPA and Personality Traits")
plt.show()
```