Data Analyst Positions Job Listing
QueryCrafters (Group7)

Cheung Hang Mang, Jeff
Wong Tat Hang, Hector

Exploratory Data Analysis (EDA) Mini Project
by Christian Action Training Services

Data Collection and Preprocessing

Data Gathering

Scraped Glassdoor job listings for 'Data Analyst Jobs' by Kaggle.

Data Cleaning

Handling missing values, outliers, and formatting data for analysis.

Data Integration

Combining data from different format into a unified dataset.

Data Transformation

Converting and preparing data for analysis and modeling.

Exploratory data analysis techniques

- **Data Cleaning:** Identify and handle missing data, outliers, and duplicates.
- **Summary Statistics:** Calculate mean, median,mode and variance for initial insights.
- **Correlation Analysis:** Explore relationships between variables using correlation coefficients.

Data Cleaning

Job Title	Salary Estimate	Job Description	Rating	Company Name	Location	Headquarters	Size	Founded	Type of ownership	Industry	Sector	Revenue	Competitors	Easy Apply
Data Analyst, Center on Immigration and Justic...	$37K-$66K (Glassdoor est.)	Are you eager to roll up your sleeves and harn...	3.2	Vera Institute of Justice\n3.2	New York, NY	New York, NY	201 to 500 employees	1961	Nonprofit Organization	Social Assistance	Non-Profit	$100 to $500 million (USD)	-1	True
Quality Data Analyst	$37K-$66K (Glassdoor est.)	Overview\n\nProvides analytical and technical ...	3.8	Visiting Nurse Service of New York\n3.8	New York, NY	New York, NY	10000+ employees	1893	Nonprofit Organization	Health Care Services & Hospitals	Health Care	$2 to $5 billion (USD)	-1	-1
Senior Data Analyst, Insights & Analytics Team...	$37K-$66K (Glassdoor est.)	We're looking for a Senior Data Analyst who ha...	3.4	Squarespace\n3.4	New York, NY	New York, NY	1001 to 5000 employees	2003	Company - Private	Internet	Information Technology	Unknown / Non-Applicable	GoDaddy	-1
Data Analyst	$37K-$66K (Glassdoor est.)	Requisition NumberRR-0001939\nRemote:Yes\nWe c...	4.1	Celerity\n4.1	New York, NY	McLean, VA	201 to 500 employees	2002	Subsidiary or Business Segment	IT Services	Information Technology	$50 to $100 million (USD)	-1	-1
Reporting Data Analyst	$37K-$66K (Glassdoor est.)	ABOUT FANDUEL GROUP\n\nFanDuel Group is a worl...	3.9	FanDuel\n3.9	New York, NY	New York, NY	501 to 1000 employees	2009	Company - Private	Sports & Recreation	Arts, Entertainment & Recreation	$100 to $500 million (USD)	DraftKings	True

Salary Estimate

The "Salary Estimate" column contains salary ranges in the format of "$37K-$66K (Glassdoor est.)"

Company Name

The "Company Name" column encounter some issues such as inconsistent formatting or special characters. ("\n"+Rating)

Rating scale

Find the best jobs by salary and company rating

Missing Values

```
missing_values = df.isnull().sum()
print(missing_values)
```

✓ 0.0s

```
Unnamed: 0           0
Job Title            0
Salary Estimate      0
Job Description      0
Rating               0
Company Name         1
Location             0
Headquarters         0
Size                 0
Founded              0
Type of ownership    0
Industry             0
Sector               0
Revenue              0
Competitors          0
Easy Apply           0
dtype: int64
```

- In this dataset, there is one missing value in the 'Company Name' column.
- After careful consideration, we have chosen to ignore this missing value for our analysis, as it constitutes only a small portion of the overall data and its absence does not significantly impact the key insights we seek to derive from the dataset.

Data Cleaning

```
# Clean the "Company Name" column
df['Company Name'] = df['Company Name'].str.split('\n').str[0]

# Remove leading and trailing whitespaces
df['Company Name'] = df['Company Name'].str.strip()
```

✓ 0.0s

```
df['Company Name'].head(20)
```

✓ 0.0s

```
0              Vera Institute of Justice\n3.2
1     Visiting Nurse Service of New York\n3.8
2                           Squarespace\n3.4
3                              Celerity\n4.1
4                               FanDuel\n3.9
5                               Point72\n3.9
6                             Two Sigma\n4.4
7                 GNY Insurance Companies\n3.7
8                                   DMGT\n4.0
9                             Riskified\n4.4
10                   NYU Langone Health\n4.0
11                                  BulbHead
12                   Montefiore Medical\n3.7
13                                Known\n3.0
14                      Advisor Group\n3.4
15                   CodeGreen Solutions\n3.6
16                              Undertone\n3.8
17                                NYSTEC\n3.8
18    Education Development Center, Inc.\n3.9
19                 Teachers Pay Teachers\n4.9
Name: Company Name, dtype: object
```

Raw Data

```
df['Company Name'].head(20)
```

✓ 0.0s

```
0              Vera Institute of Justice
1     Visiting Nurse Service of New York
2                           Squarespace
3                              Celerity
4                               FanDuel
5                               Point72
6                             Two Sigma
7                 GNY Insurance Companies
8                                   DMGT
9                             Riskified
10                   NYU Langone Health
11                                  BulbHead
12                   Montefiore Medical
13                                Known
14                      Advisor Group
15                   CodeGreen Solutions
16                              Undertone
17                                NYSTEC
18    Education Development Center, Inc.
19                 Teachers Pay Teachers
Name: Company Name, dtype: object
```

After Cleaning

Splitting the "Company Name" column based on the "\n" character and selecting the first part of the split, then removing any leading or trailing whitespaces from each element in the "Company Name" column.

Data visualization methods

1. **Pandas**

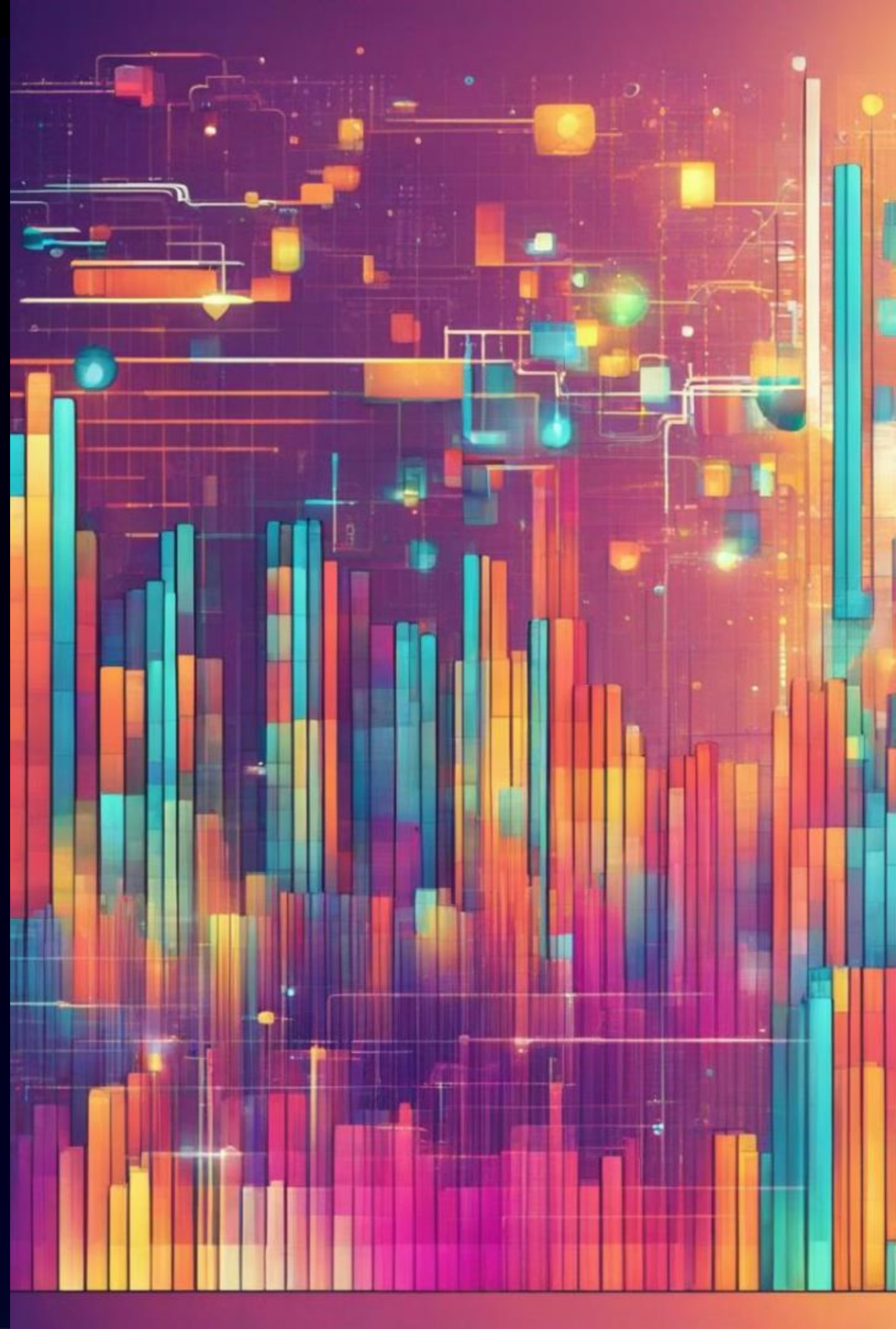
   It provides a simple interface to create common plots such as line plots, bar plots, and scatter plots directly from pandas DataFrames

2. **Matplotlib**

   It provides a wide range of plot types, including line plots, scatter plots, bar plots, histograms, and many more.

3. **Seaborn**

   Simplifies the process of creating complex visualizations such as heatmaps, violin plots, box plots, and regression plots.

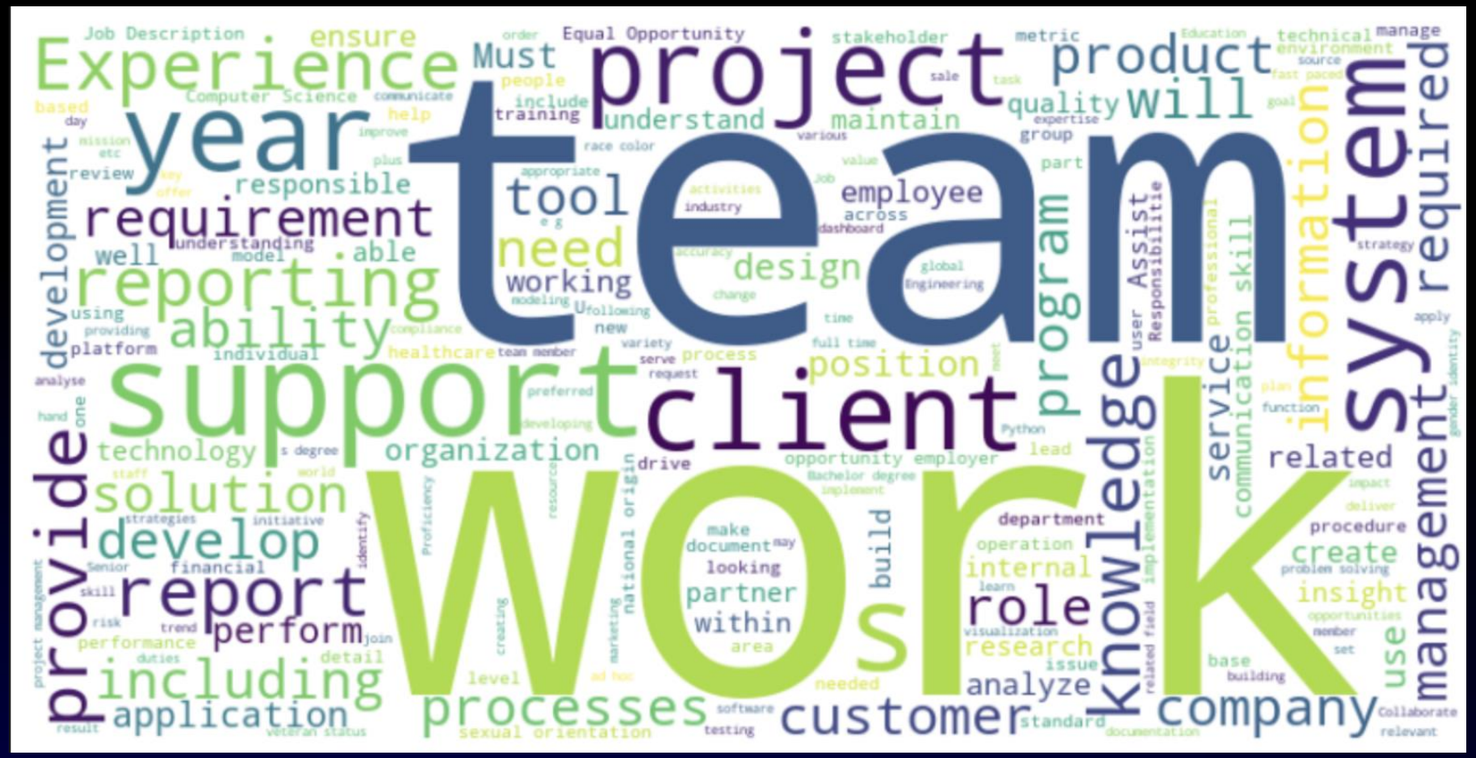


Most Frequency Words

```
###WordCloud###

# Concatenate the two columns into a single string
text = ' '.join(df['Job Title']) + ' ' + ' '.join(df['Job Description'])

# Tokenize the text into individual words
tokens = word_tokenize(text)

# Define the keywords you want to drop
unwanted_keywords = ["data", "analyst", "experience", "analysis", "Data Analyst", "Ability", "business", "analytic" ]  # Add your unwanted keywords here

# Remove the unwanted keywords from the text
for keyword in unwanted_keywords:
    text = text.replace(keyword, "")

# Remove stopwords and common verbs
stop_words = set(stopwords.words('english'))
common_verbs = ['is', 'are', 'was', 'were', 'be', 'been', 'am', 'being', 'do', 'does', 'did', 'doing', 'have', 'has', 'had', 'having']
filtered_tokens = [word for word in tokens if word.lower() not in stop_words and word.lower() not in common_verbs]

# Join the filtered tokens back into a single string
filtered_text = ' '.join(filtered_tokens)

# Generate the word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

# Display the word cloud using matplotlib
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

✓ 4.5s

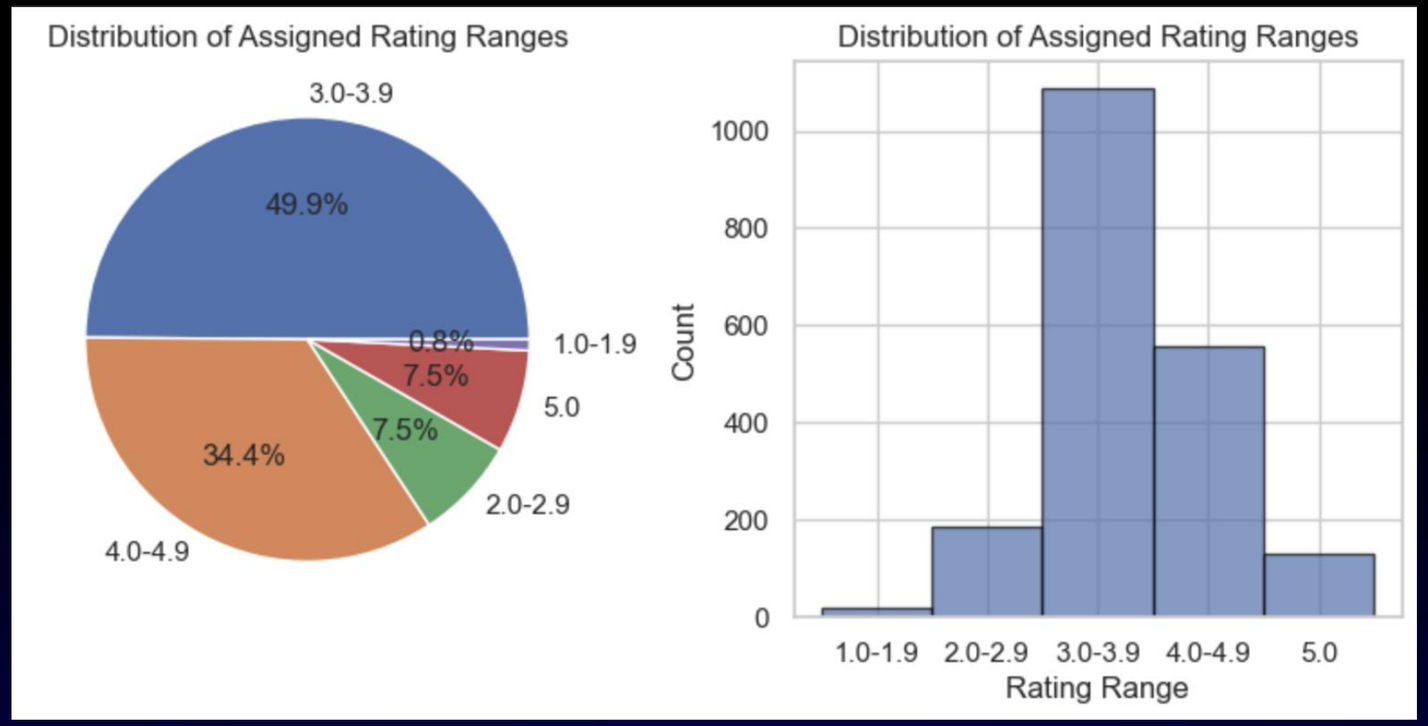
Most Frequency Words



Represents the frequency of words in a text corpus by displaying them as different-sized words in a cloud-like arrangement, where the size of each word corresponds to frequency.

Based on the words of mentioned, it appears that the text data you used for generating the word cloud is related to a context where teamwork, client interaction, support, and projects are important. These words likely have higher frequencies in the text, indicating their significance or prominence.

Distribution Assigned Rating

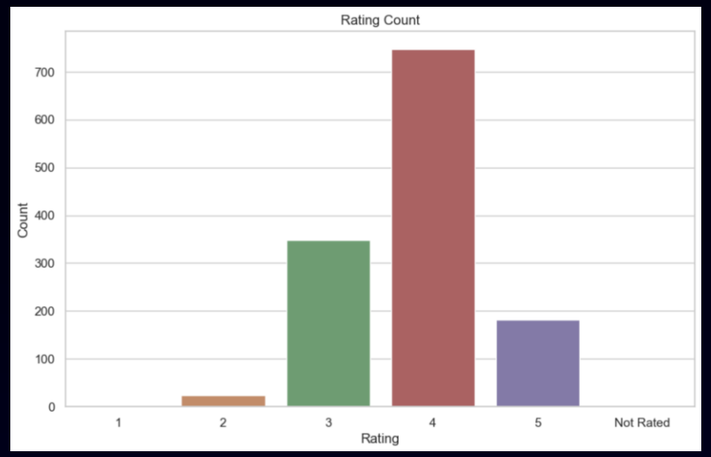


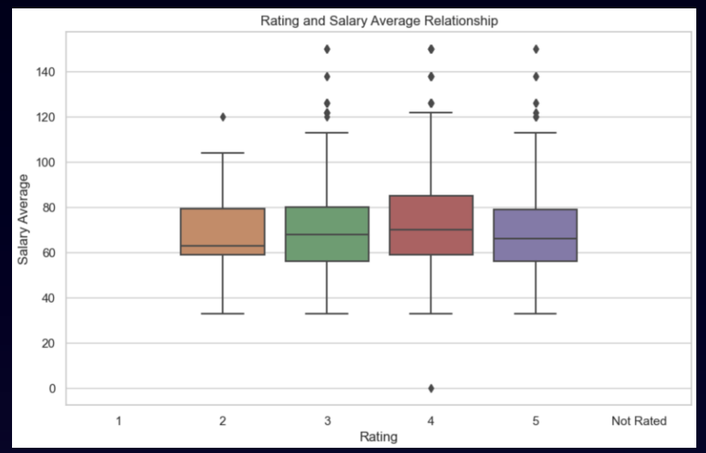
The histogram displays the frequency distribution of the assigned rating ranges, illustrating the count of ratings falling within each range category.

The pie chart visually represents the distribution of assigned rating ranges in the dataset, showing the percentage of each rating range category.

```
###Pie chart and histogram

# Define the bin edges and labels
bin_edges = [0, 1.9, 2.9, 3.9, 4.9, 5.1]  # The edges include the lower boundary and exclude the upper boundary
bin_labels = ['1.0-1.9', '2.0-2.9', '3.0-3.9', '4.0-4.9', '5.0']

# Create a new column with the assigned ranges
df['Rating Range'] = pd.cut(df['Rating'], bins=bin_edges, labels=bin_labels, right=False)

# Count the frequency of each assigned rating range
rating_range_counts = df['Rating Range'].value_counts()

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 4))

# Plot the pie chart on the first subplot
ax1.pie(rating_range_counts, labels=rating_range_counts.index, autopct='%1.1f%%')
ax1.set_title('Distribution of Assigned Rating Ranges')

# Plot the histogram on the second subplot
ax2.hist(df['Rating'], bins=[1, 2, 3, 4, 5, 6], edgecolor='black', alpha=0.7)
ax2.set_xticks([1.5, 2.5, 3.5, 4.5, 5.5])
ax2.set_xticklabels(['1.0-1.9', '2.0-2.9', '3.0-3.9', '4.0-4.9', '5.0'])
ax2.set_xlabel('Rating Range')
ax2.set_ylabel('Count')
ax2.set_title('Distribution of Assigned Rating Ranges')

# Adjust spacing between subplots
plt.subplots_adjust(wspace=0.3)

# Display the combined chart
plt.show()
```

Rating and Salary Average Relationship






```
###Countplot and Boxplot

# Create a copy of the DataFrame
df_filtered = df.copy()

# Replace -1 with 'Not Rated' in the 'Rating_round' column
df_filtered['Rating_round'] = df_filtered['Rating'].apply(lambda x: 'Not Rated' if x == -1 else round(x))

# Define the desired order of ratings
rating_order = [ 1, 2, 3, 4, 5,'Not Rated']

# Group the data by rounded ratings and calculate the mean salary and count
df_grouped = df_filtered.groupby('Rating_round').agg({'Salary Average': 'mean', 'Rating': 'count'}).reset_index()
df_grouped = df_grouped.rename(columns={'Rating': 'Rating Count'})

# Plot count of each rating
plt.figure(figsize=(10, 6))
sns.countplot(x=df_filtered['Rating_round'], order=rating_order)
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Rating Count')
plt.show()

# Plot relationship between rating and salary average
plt.figure(figsize=(10, 6))
sns.boxplot(x=df_filtered['Rating_round'], y=df_filtered['Salary Average'], order=rating_order)
plt.xlabel('Rating')
plt.ylabel('Salary Average')
plt.title('Rating and Salary Average Relationship')
plt.show()
```

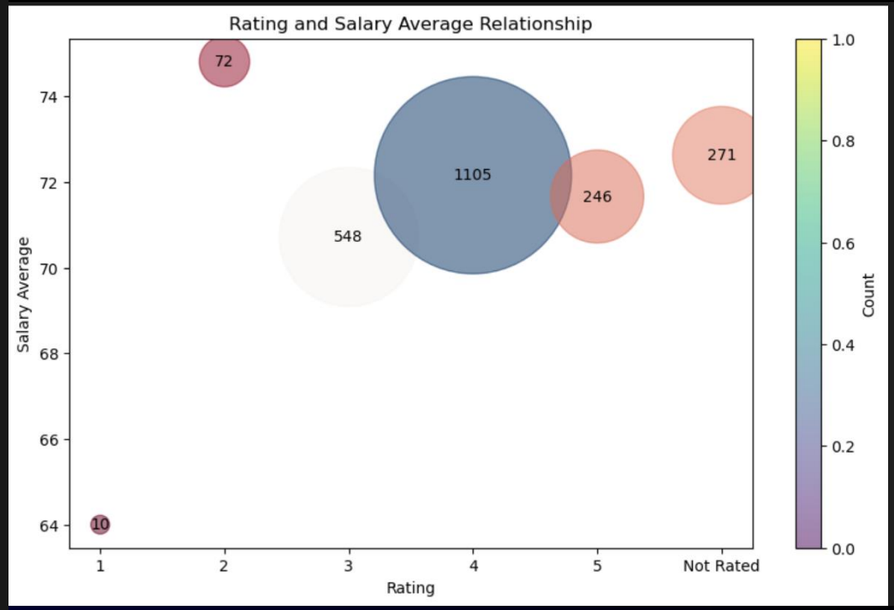
✓ 0.1s

A boxplot visually depicts the relationship between rating categories and average salaries, revealing variations or trends in salary distribution. A bar chart displays the count of each rating category, giving an overview of the distribution of ratings in the dataset.

Rating and Salary Relationship (Bubble plot)

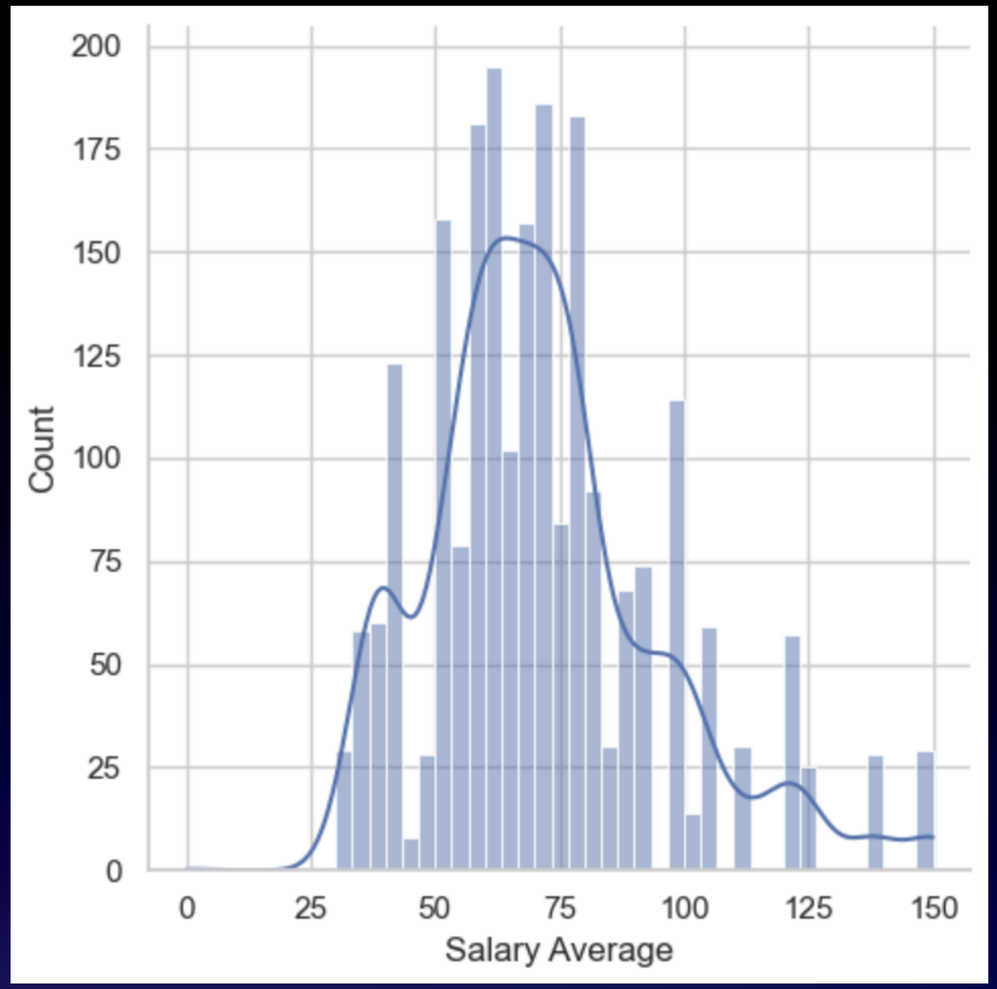
```python
###Bubble plot

df_grouped['Rating_round'] = df_grouped['Rating_round'].replace(-1.0, 'Not Rated')  # Replace -1.0 with 'Not Rated' in Rating_round column
df_grouped['Rating_round'] = df_grouped['Rating_round'].fillna('Not Rated').astype(str)  # Fill missing values with 'Not Rated' and convert to string
df_grouped = df_grouped.rename(columns={'Rating': 'Rating Count'})  # Rename 'Rating' column to 'Rating Count'

cmap = cm.get_cmap('RdBu')  # Get colormap
norm = Normalize(vmin=df_grouped['Rating Count'].min(), vmax=df_grouped['Rating Count'].max())  # Normalize the Rating Count values

fig, ax = plt.subplots(figsize=(10, 6))  # Create a figure and axis

# Scatter plot
sc = ax.scatter(df_grouped['Rating_round'], df_grouped['Salary Average'], s=df_grouped['Rating Count']*15, alpha=0.5, c=cmap(norm(df_grouped['Rating Count'])))
# Set x-axis label
ax.set_xlabel('Rating')
# Set y-axis label
ax.set_ylabel('Salary Average')
# Set plot title
ax.set_title('Relationship between Rating and Salary Average')

# Add text labels for each data point
for i in range(df_grouped.shape[0]):
    ax.text(df_grouped['Rating_round'].iloc[i], df_grouped['Salary Average'].iloc[i], df_grouped['Rating Count'].iloc[i], ha='center', va='center', color='black')

# Add colorbar
fig.colorbar(sc, ax=ax, label='Count', cmap='RdBu')

# Display the plot
plt.show()
```

✓ 0.1s



This bubble plot showcases the relationship between the average salary and rating count for different categories, with bubble sizes indicating the rating count and colors representing the normalized rating count proportion

Salary Average Kernel density plot (KDE)

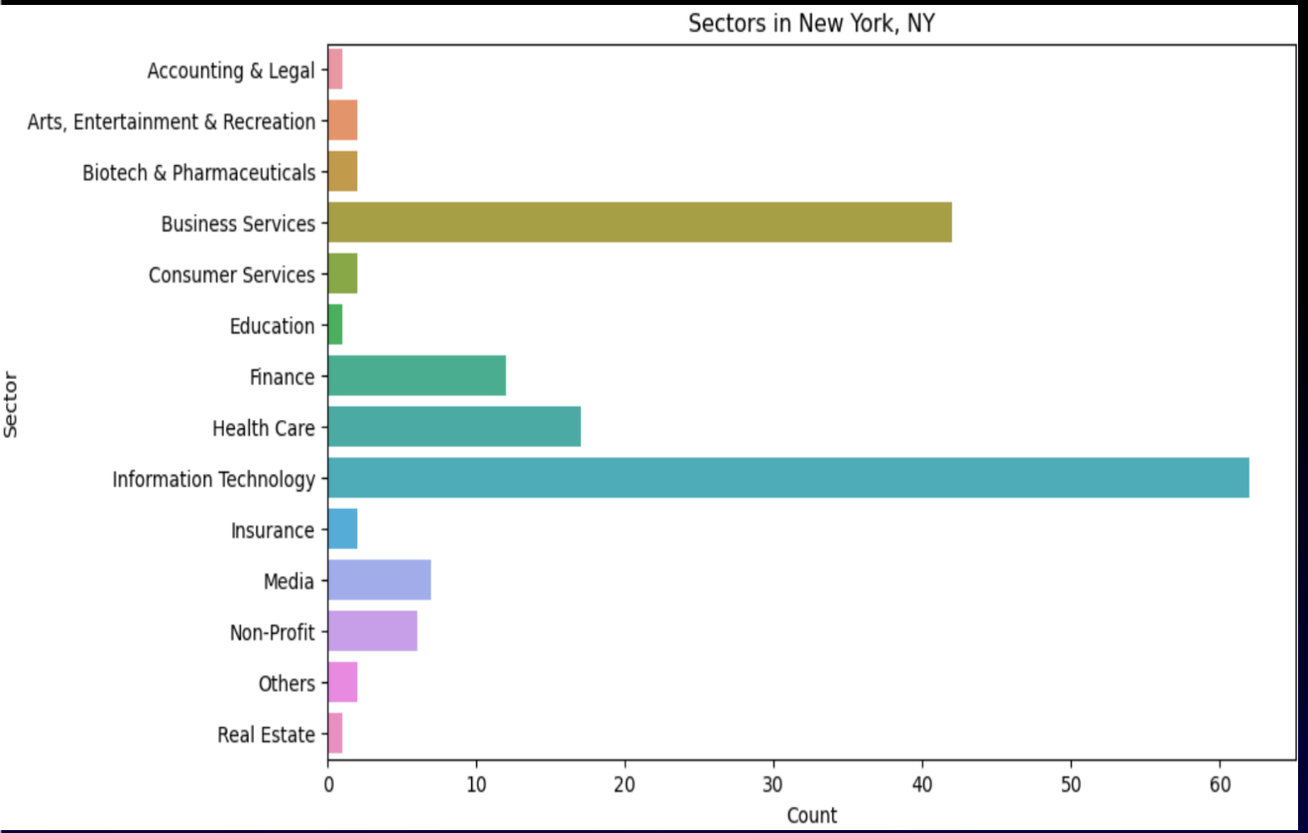



```python
###Kernel density plot with 'Salary Average'

# Convert infinite values to NaN
df.replace([np.inf, -np.inf], np.nan, inplace=True)

# Plot the distribution of 'Salary Average' with a kernel density estimate
sns.displot(data=df, x='Salary Average', kde=True)

# Display the plot
plt.show()
```

KDE use normal distribution provides a smooth and continuous estimation of the probability density function, allowing us to visualize the distribution of the 'Salary Average' variable in our dataset.

KDE plot of the 'Salary Average' variable provides a visual representation of its distribution, allowing us to observe the concentration and spread of salary averages within the dataset.

Max Sector Location (Bar Chart)




```python
###Bar chart with plot_sector

#Defines the function and calls it with the argument 3 to generate the bar plot for the desired rank

def plot_sector_counts(rank):
    # Replace -1 with 'Others' in the 'Sector' column
    df['Sector'] = df['Sector'].replace("-1", 'Others')

    sector_counts = df.groupby(['Location', 'Sector']).size().reset_index(name='Counts')
    location_counts = sector_counts.groupby('Location')['Counts'].sum().reset_index()

    sorted_location_counts = location_counts.sort_values(by='Counts', ascending=False)

    nth_max_sector_location = sorted_location_counts['Location'].iloc[rank-1]

    nth_max_sector_location_data = sector_counts[sector_counts['Location'] == nth_max_sector_location]

    plt.figure(figsize=(10, 6))
    sns.barplot(x='Counts', y='Sector', data=nth_max_sector_location_data, errcolor='black', errwidth=0)
    plt.xlabel('Count')
    plt.ylabel('Sector')
    plt.title(f'Sectors in {nth_max_sector_location}')

    plt.show()

plot_sector_counts(1)
```

✓ 0.1s

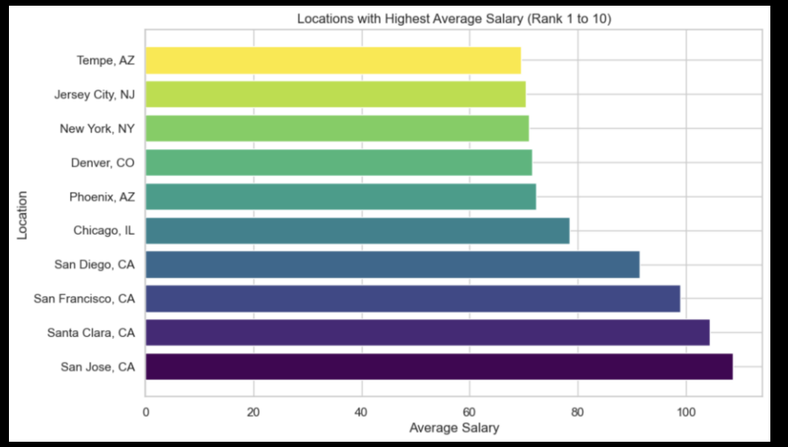
Display jobs across various industries for a specific location

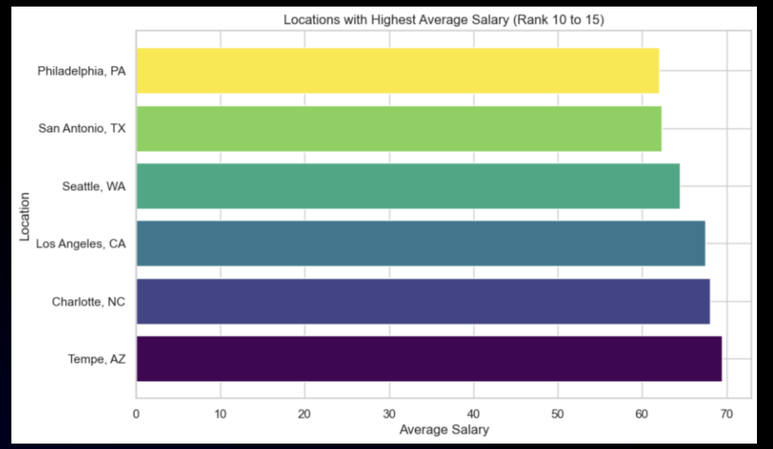
(Customizable, from the area with the most to the least job vacancies)

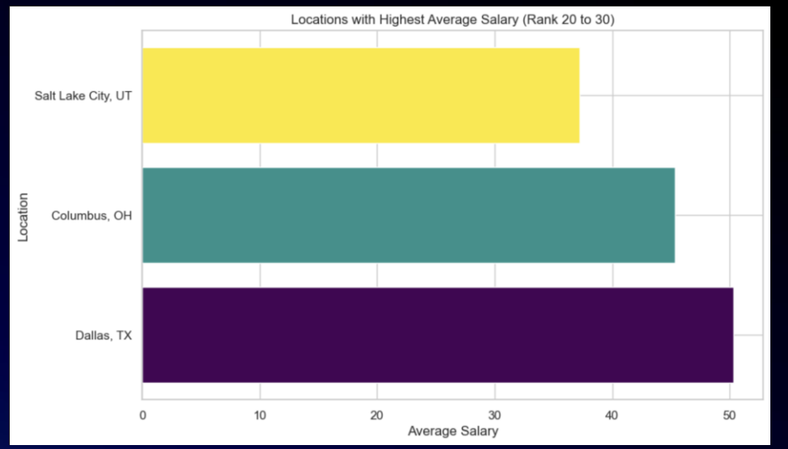
Then plots a horizontal bar chart automatically

Highest Location Salary

```python
###Horizontal bar chart

def plot_average_salaries(start_rank, end_rank):
    # Calculate the sample counts for each location
    location_counts = df['Location'].value_counts().reset_index()
    location_counts.columns = ['Location', 'Count']

    # Filter locations with sample counts greater than the specified threshold
    locations_with_enough_samples = location_counts[location_counts['Count'] > 20]['Location']

    # Calculate the average salary for each location
    average_salary_by_location = df[df['Location'].isin(locations_with_enough_samples)].groupby('Location')['Salary Average'].mean().reset_index()

    # Sort locations based on average salary
    sorted_average_salary_by_location = average_salary_by_location.sort_values(by='Salary Average', ascending=False)

    # Select the data for the specified rank range
    average_salaries_in_rank_range = sorted_average_salary_by_location.iloc[start_rank-1:end_rank]

    # Generate colors for the bars
    num_bars = len(average_salaries_in_rank_range)
    cmap = plt.get_cmap('viridis')
    colors = cmap(np.linspace(0, 1, num_bars))

    # Plot bar chart using matplotlib
    plt.figure(figsize=(10, 6))
    plt.barh(average_salaries_in_rank_range['Location'], average_salaries_in_rank_range['Salary Average'], color=colors)
    plt.xlabel('Average Salary')
    plt.ylabel('Location')
    plt.title(f'Locations with Highest Average Salary (Rank {start_rank} to {end_rank})')
    plt.show()

plot_average_salaries(1, 10)
plot_average_salaries(10, 15)
plot_average_salaries(20, 30)
```





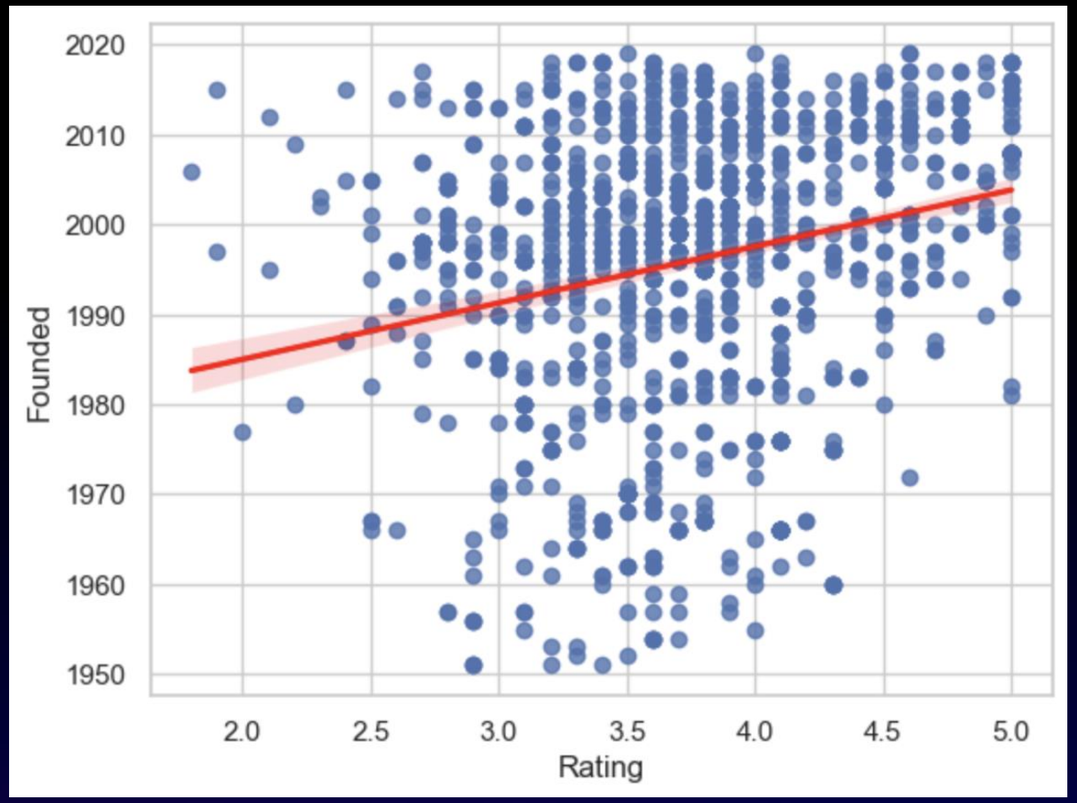


Display the average salaries of different locations,

Easy comparison locations with the highest average salaries

(Also customizable plot chart automatically)

Scatter plot with regression line




```
###Scatter plot with a regression line

# Convert 'Rating' column to numeric, replacing any non-numeric values with NaN
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')

# Convert 'Founded' column to numeric, replacing any non-numeric values with NaN
df['Founded'] = pd.to_numeric(df['Founded'], errors='coerce')

# Filter out data where 'Founded' is -1 and keep only data where 'Founded' is after 1950
df = df[(df['Founded'] != -1) & (df['Rating'] != -1) & (df['Founded'] > 1950)]

# Create a scatter plot and add a regression line
sns.regplot(x='Rating', y='Founded', data=df, line_kws={"color": "red"})
plt.show()
```

✓ 0.1s

Relationship between Ratings and Founding years of companies,
allowing us to observe any potential patterns, trends, or correlations between them.

Results and insights

The thorough analysis of the dataset has uncovered significant correlations between variables, unveiling previously unknown patterns and relationships. These insights hold great value in informing decision-making processes and deepening our understanding of the underlying mechanisms.

To enhance the clarity of the results, visually engaging representations have been developed. These visual insights offer a comprehensive understanding of the findings, presenting the data analysis results in a vibrant and dynamic manner.

A comparative analysis of the data across different job titles, industries, or company sizes can provide valuable insights into variations and differences in salaries, job descriptions, ratings, and other relevant factors. This analysis can shed light on the distinctive characteristics and requirements of different roles or sectors.

Limitation

Time Limit： We can do sorting function to find best jobs by salary and company rating to combine previous two functions.

System compatibility issue(Mac & Windows): The colors in the illustrated image are different.

GitHub Newbie

Reference

Data Analyst Jobs - Kaggle

https://www.kaggle.com/datasets/andrewmvd/data-analyst-jobs

Thanks for watching!