t

Data Analyst Job Market Analysis & Salary Prediction Project

Data Collection

```
import pandas as pd


data = pd.read_csv(r"C:\Users\NILAM\Desktop\DataAnalyst.csv")
pd.set_option('display.max_columns', 4)
pd.set_option('display.width', 200)
print(data.head())
```

```
   Unnamed: 0                                          Job Title  ... Competitors Easy Apply
0           0  Data Analyst, Center on Immigration and Justic...  ...          -1       TRUE
1           1                                Quality Data Analyst  ...          -1         -1
2           2  Senior Data Analyst, Insights & Analytics Team...  ...      GoDaddy         -1
3           3                                       Data Analyst  ...          -1         -1
4           4                             Reporting Data Analyst  ...  DraftKings       TRUE
```

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2253 entries, 0 to 2252
Data columns (total 16 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         2253 non-null   int64
 1   Job Title          2253 non-null   object
 2   Salary Estimate    2253 non-null   object
 3   Job Description    2253 non-null   object
 4   Rating             2253 non-null   float64
 5   Company Name       2252 non-null   object
 6   Location           2253 non-null   object
 7   Headquarters       2253 non-null   object
 8   Size               2253 non-null   object
 9   Founded            2253 non-null   int64
 10  Type of ownership  2253 non-null   object
 11  Industry           2253 non-null   object
 12  Sector             2253 non-null   object
 13  Revenue            2253 non-null   object
 14  Competitors        2253 non-null   object
 15  Easy Apply         2253 non-null   object
dtypes: float64(1), int64(2), object(13)
memory usage: 281.8+ KB
None
```

Exploratory Data Analysis (EDA)

```
pd.set_option('display.max_columns',  6)
pd.set_option('display.width',  100)
print(f"Duplicate rows: {data.duplicated().sum()}")
print(data.describe(include='all'))
```

```
Duplicate rows: 0
        Unnamed: 0     Job Title              Salary  Estimate  ...
Revenue  \
count     2253.0000          2253                        2253  ...
2253
unique          NaN          1272                          90  ...
14
top             NaN  Data Analyst  $42K-$76K (Glassdoor est.)  ...  Unknown / Non-
Applicable
freq            NaN           405                          57  ...
615
mean      1126.0000           NaN                         NaN  ...
NaN
std        650.5294           NaN                         NaN  ...
NaN
min          0.0000           NaN                         NaN  ...
NaN
25%        563.0000           NaN                         NaN  ...
NaN
50%       1126.0000           NaN                         NaN  ...
NaN
75%       1689.0000           NaN                         NaN  ...
NaN
max       2252.0000           NaN                         NaN  ...
NaN



        Competitors Easy  Apply
count           2253        2253
unique           291           2
top               -1          -1
freq            1732        2173
mean             NaN         NaN
std              NaN         NaN
min              NaN         NaN
25%              NaN         NaN
50%              NaN         NaN
75%              NaN         NaN
max              NaN         NaN
```

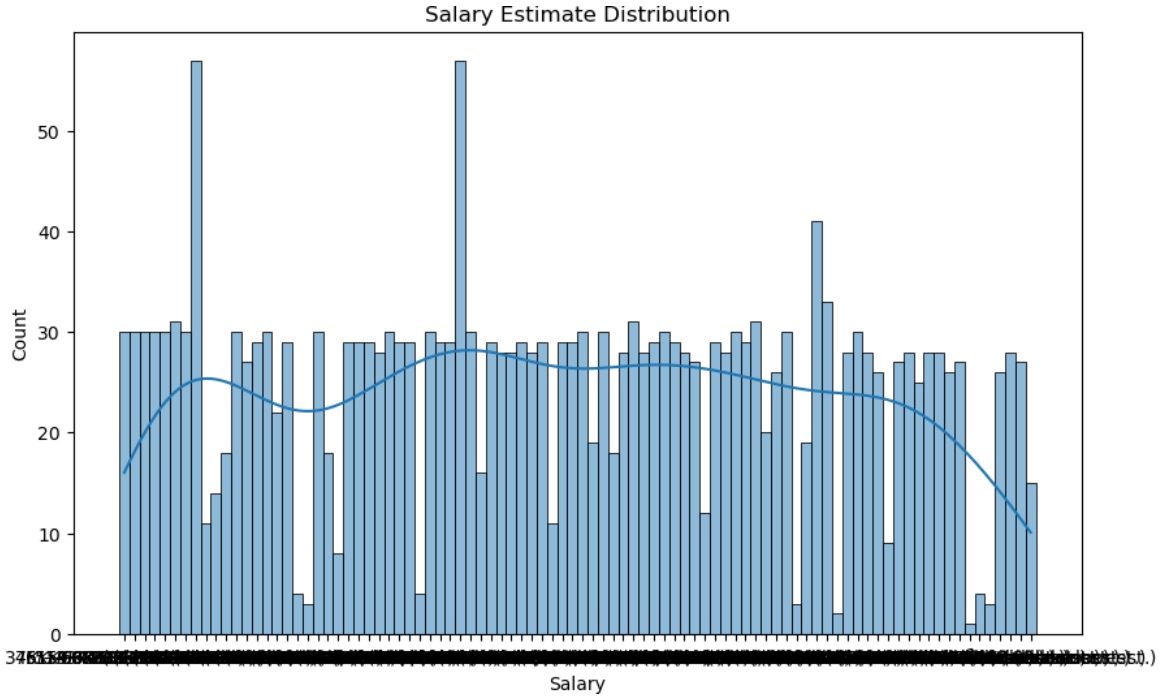
```
for col in ['Job Title', 'Type of ownership', 'Industry','Sector']:
  print(data[col].value_counts().head())
```

```
Job Title
Data Analyst              405
Senior Data Analyst        90
Junior Data Analyst        30
Business Data Analyst      28
Sr. Data Analyst           21
Name: count, dtype: int64
Type of ownership
Company - Private                  1273
Company - Public                    452
-1                                  163
Nonprofit Organization              124
Subsidiary or Business Segment       89
Name: count, dtype: int64
Industry
-1                                    353
IT Services                           325
Staffing & Outsourcing                323
Health Care Services & Hospitals      151
Computer Hardware & Software          111
Name: count, dtype: int64
Sector
Information Technology     570
Business Services          524
-1                         353
Finance                    169
Health Care                151
Name: count, dtype: int64
```

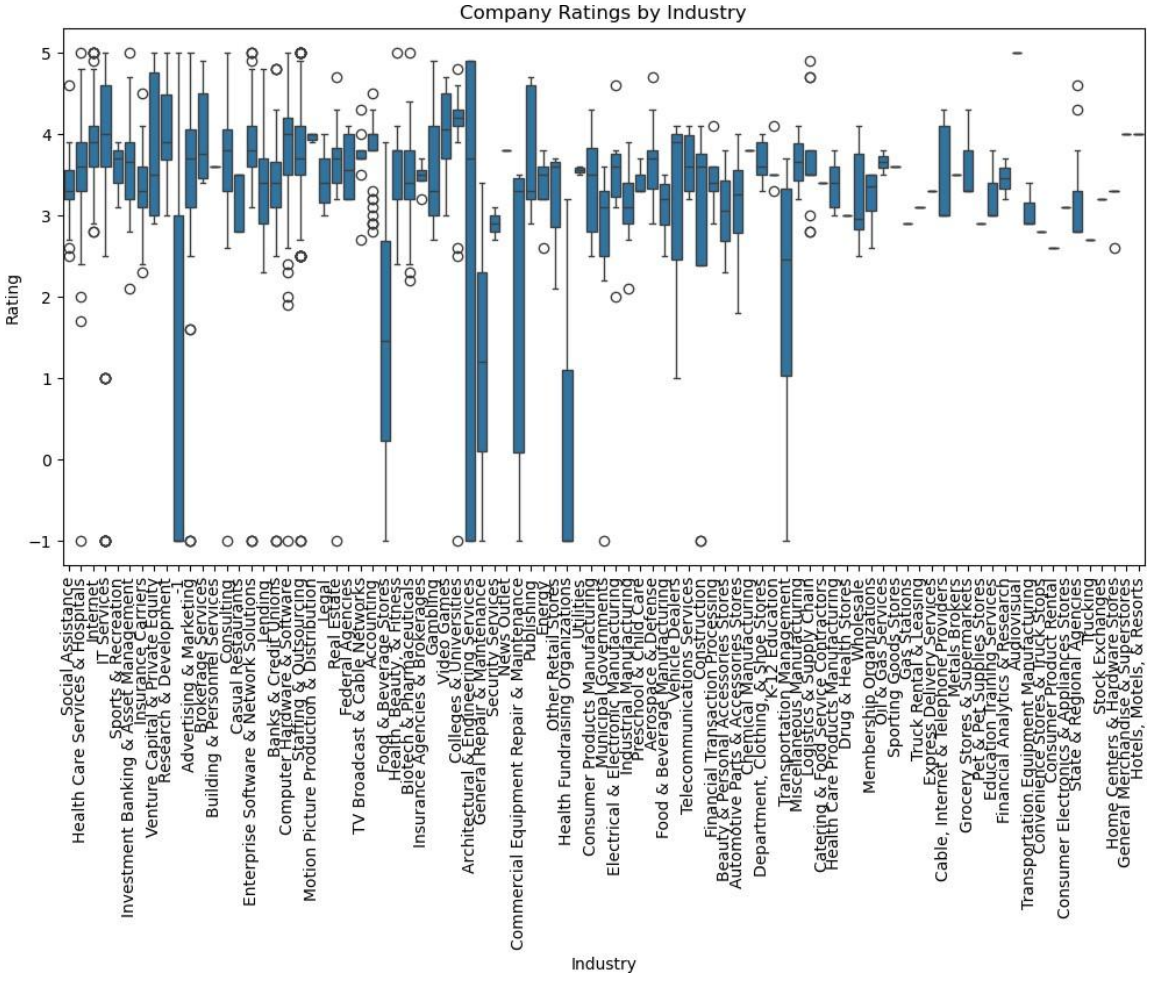
Visualization

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.histplot(data['Salary Estimate'], kde=True, bins=20)
plt.title("Salary Estimate Distribution")
plt.xlabel("Salary")
plt.show()
```

t




```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Industry', y='Rating', data=data)
plt.xticks(rotation=90)
plt.title("Company Ratings by Industry")
plt.show()
```



Data Cleaning

```
print(data.isnull().sum())
data['Rating'].fillna(data['Rating'].median(), inplace=True)
threshold = len(data) * 0.3
data = data.dropna(thresh=threshold, axis=1)
categorical_cols = ['Company Name', 'Industry', 'Sector', 'Type of ownership']
data[categorical_cols] = data[categorical_cols].fillna(method='ffill')
```

```
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
```

```
data.columns = data.columns.str.strip().str.replace('\u00a0', ' ').str.lower()
```

Feature Engineering

```
import pandas as pd
import re
data = pd.read_csv(r"C:\Users\NILAM\Desktop\DataAnalyst.csv")
salary_col = None
for col in data.columns:
    if "salary" in col.lower():
        salary_col = col
        break
if salary_col is None:
    raise ValueError("No salary-like column found in dataset.")
```

```
def parse_salary(s):
    if pd.isna(s):
        return (None, None, None, False)
    s_low = str(s).lower()
    is_hourly = bool(re.search(r'per hour|/hour|per hr|hourly|/hr', s_low))
    s_clean = re.sub(r'\(.*?\)', '', s_low)
    s_clean = re.sub(r'employer provided salary:|employer provided|glassdoor est
    s_clean = s_clean.replace(',', ' ').replace('$', ' ')
    s_clean = re.sub(r'per year|per yr|per annum|per month|per mo|yearly|annum|/
    matches = re.findall(r'(\d+(?:\.\d+)?)(k)?', s_clean)
```

```
    if not matches:
        return (None, None, None, is_hourly)
    nums = []
    had_k_flag = False
    for num, kflag in matches:
        val = float(num)
        if kflag:
            val *= 1000.0
            had_k_flag = True
        nums.append(val)
    if len(nums) == 1:
        min_val = max_val = nums[0]
    else:
        min_val = nums[0]
        max_val = nums[1]
    if max_val is not None and max_val < 1000 and not had_k_flag and not is_hour
        min_val *= 1000.0
        max_val *= 1000.0
    if is_hourly:
        min_val = min_val * 2080.0
        max_val = max_val * 2080.0
    avg = (min_val + max_val) / 2.0 if (min_val is not None and max_val is not N
    return (min_val, max_val, avg, is_hourly)
```

```
parsed = data[salary_col].apply(parse_salary).apply(pd.Series)
parsed.columns = ['Min_Annual', 'Max_Annual', 'Avg_Annual', 'Is_Hourly']
data = pd.concat([data, parsed], axis=1)
data.drop(salary_col, axis=1, inplace=True)
data['Min_K'] = data['Min_Annual'] / 1000
data['Avg_K'] = data['Avg_Annual'] / 1000
```

```
print(data.columns.tolist())
```

```
['Unnamed: 0', 'Job Title', 'Job Description', 'Rating', 'Company Name', 'Locatio
n', 'Headquarters', 'Size', 'Founded', 'Type of ownership', 'Industry', 'Sector',
'Revenue', 'Competitors', 'Easy Apply', 'Min_Annual', 'Max_Annual', 'Avg_Annual',
'Is_Hourly', 'Min_K', 'Avg_K']
```

```
data['Max_K'] = data['Max_Annual'] / 1000
```

```
print(data[['Job Title', 'Min_K', 'Avg_K', 'Max_K']].head())
```

```
                                           Job Title  Min_K  Avg_K  Max_K
0  Data Analyst, Center on Immigration and Justic...   37.0   51.5   66.0
1                                Quality Data Analyst   37.0   51.5   66.0
2  Senior Data Analyst, Insights & Analytics Team...   37.0   51.5   66.0
3                                     Data Analyst   37.0   51.5   66.0
4                              Reporting Data Analyst   37.0   51.5   66.0
```

```
data['Python'] = data['Job Description'].str.contains('Python',case=False, na=Fa
data['Excel'] = data['Job Description'].str.contains('Excel',case=False, na=Fals
data['Tech_Skills'] = data['Python'] + data['Excel']
```

```
import pandas as pd
if 'Location' not in data.columns:
    raise KeyError("No 'Location' column found in dataframe.")
loc = data['Location'].fillna('').astype(str)
loc_clean = loc.str.replace(r'\(.*?\)', '', regex=True)
loc_clean = loc_clean.str.replace(r'\s*-\s*', ',', regex=True)
loc_clean = loc_clean.str.replace(r'\s*,\s*', ',', regex=True).str.strip(' ,')
```

t

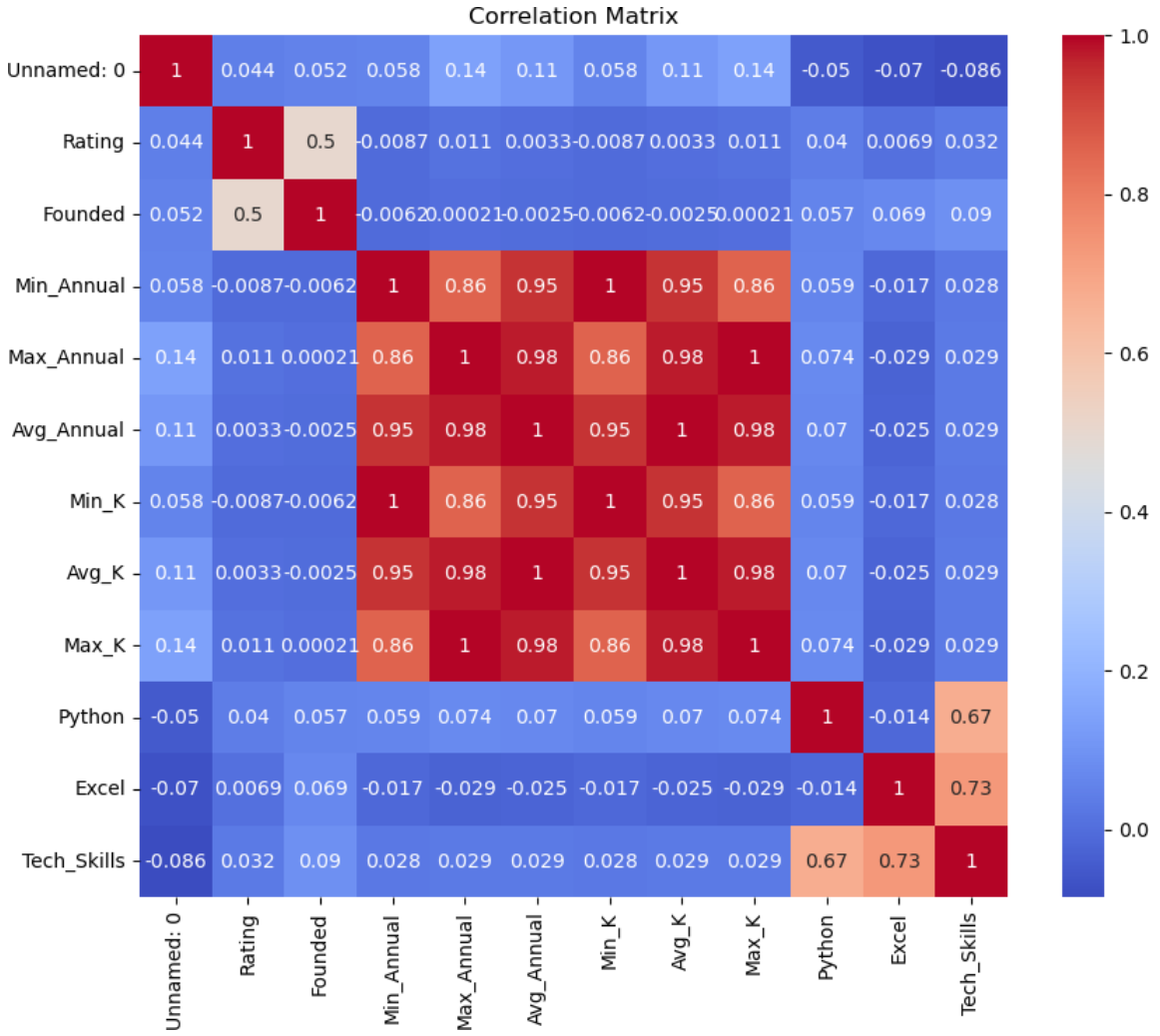
```
parts = loc_clean.str.split(',', n=1, expand=True)
data['City'] = parts[0].str.strip()
if parts.shape[1] > 1:
    data['State_raw'] = parts[1].str.strip()
else:
    data['State_raw'] = pd.NA

data['State'] = data['State_raw'].fillna('').str.split().str[0].str.replace(r'[^
data.loc[data['City'].str.strip() == '', 'City'] = pd.NA
data.loc[data['State'] == '', 'State'] = pd.NA
remote_keywords = {'remote', 'work from home', 'wfh', 'anywhere'}
remote_mask = data['City'].fillna('').str.lower().isin(remote_keywords)
data.loc[remote_mask, ['City','State']] = pd.NA
data.drop(columns=['State_raw'],  inplace=True)
print(data[['Location','City','State']].head(20))
```

	Location	City	State
0	New York, NY	New York	NY
1	New York, NY	New York	NY
2	New York, NY	New York	NY
3	New York, NY	New York	NY
4	New York, NY	New York	NY
5	New York, NY	New York	NY
6	New York, NY	New York	NY
7	New York, NY	New York	NY
8	New York, NY	New York	NY
9	New York, NY	New York	NY
10	New York, NY	New York	NY
11	Fairfield, NJ	Fairfield	NJ
12	New York, NY	New York	NY
13	New York, NY	New York	NY
14	Jersey City, NJ	Jersey City	NJ
15	New York, NY	New York	NY
16	New York, NY	New York	NY
17	New York, NY	New York	NY
18	New York, NY	New York	NY
19	New York, NY	New York	NY

Statistics

```
import matplotlib.pyplot as plt
import seaborn as sns
numeric_data = data.select_dtypes(include=['number'])
plt.figure(figsize=(10, 8))
sns.heatmap(numeric_data.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```

t




```
y = data['Avg_Annual']
y = data['Avg_K']
```

Model Development

```
from sklearn.model_selection import train_test_split
features = ['Rating', 'Tech_Skills', 'Size', 'Founded']
X = data[features]
y = data['Avg_Annual']
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.2, random_s
```

```
from sklearn.preprocessing import LabelEncoder
df = data.copy()
categorical_cols = ['Size']
le = LabelEncoder()
for col in categorical_cols:
    df[col] = le.fit_transform(df[col].astype(str))
features = ['Rating', 'Size', 'Founded']
X = df[features]
y = df['Avg_Annual']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
model = RandomForestRegressor(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"MAE: {mae}, R2 Score: {r2}")
```

```
MAE: 21180.875443502526, R2 Score: -0.2512699883838907
```

```
import streamlit as st
features = ['Rating', 'Tech_Skills', 'Size', 'Founded']
X = df[features]
y = df['Avg_Annual']
rating = st.slider("Company Rating", 1, 5, 3)
size = st.selectbox("Company Size", [0, 1, 2])
founded = st.number_input("Year Founded", min_value=1900, max_value=2023, value=

prediction = model.predict([[rating, size, founded]])
st.write(f"Predicted Salary: ${prediction[0]:,.2f}")
```

Dataset Overview

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import re
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
warnings.filterwarnings('ignore')
```

```
data_analyst_jobs = pd.read_csv(r'C:\Users\NILAM\Desktop\DataAnalyst.csv')
```

```
data_analyst_jobs = data_analyst_jobs.drop('Unnamed: 0',axis=1)
data_analyst_jobs = data_analyst_jobs.drop('Founded', axis=1)
data_analyst_jobs = data_analyst_jobs.drop('Competitors',axis=1)
print(f'Number of rows:{data_analyst_jobs.shape[0]};Number of columns:{data_anal
No of missing values:{sum(data_analyst_jobs.isna().sum())}')
```

```
Number of rows:2253;Number of columns:13;No of missing values:1
```

```
data_analyst_jobs.head()
```

t

	Job Title	Salary Estimate	Job Description	Rating	Company Name	Location
0	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
1	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
2	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
3	Data Analyst	37K–66K (Glassdoor est.)	Requisition NumberRR-0001939\nRemote:Yes\nWe c...	4.1	Celerity\n4.1	New York, NY
4	Reporting Data Analyst	37K–66K (Glassdoor est.)	ABOUT FANDUEL GROUP\n\nFanDuel Group is a worl...	3.9	FanDuel\n3.9	New York, NY

```
data_analyst_jobs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2253 entries, 0 to 2252
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Job Title          2253 non-null   object 
 1   Salary Estimate    2253 non-null   object 
 2   Job Description    2253 non-null   object 
 3   Rating             2253 non-null   float64
 4   Company Name       2252 non-null   object 
 5   Location           2253 non-null   object 
 6   Headquarters       2253 non-null   object 
 7   Size               2253 non-null   object 
 8   Type of ownership  2253 non-null   object 
 9   Industry           2253 non-null   object 
 10  Sector             2253 non-null   object 
 11  Revenue            2253 non-null   object 
 12  Easy Apply         2253 non-null   object 
dtypes: float64(1), object(12)
memory usage: 228.9+ KB
```

Renaming Columns

```
data_analyst_jobs.rename(columns={"Job Title": "job_title"},inplace=True)
data_analyst_jobs.rename(columns={"Salary Estimate":"salary_estimate"}, inplace=
data_analyst_jobs.rename(columns={"Job Description":"job_description"}, inplace=
data_analyst_jobs.rename(columns={"Company Name":"company_name"}, inplace=True)
```

```
data_analyst_jobs.rename(columns={"Location":  "location"},inplace=True)
data_analyst_jobs.rename(columns={"Headquarters":"headquarters"}, inplace=True)
data_analyst_jobs.rename(columns={"Size": "size"},inplace=True)
data_analyst_jobs.rename(columns={"Type of ownership":"type_of_ownership"}, inpl
data_analyst_jobs.rename(columns={"Industry": "industry"},inplace=True)
data_analyst_jobs.rename(columns={"Sector":"sector"},inplace=True)
data_analyst_jobs.rename(columns={"Revenue":"revenue"},inplace=True)
data_analyst_jobs.rename(columns={"EasyApply":"easy_apply"},inplace=True)
```

```
data_analyst_jobs.head()
```

	job_title	salary_estimate	job_description	Rating	company_name	loc
0	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	Yor
1	Quality Data Analyst	37K–66K (Glassdoor est.)	Overview\n\nProvides analytical and technical ...	3.8	Visiting Nurse Service of New York\n3.8	Yor
2	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	Yor
3	Data Analyst	37K–66K (Glassdoor est.)	Requisition NumberRR-0001939\nRemote:Yes\nWe c...	4.1	Celerity\n4.1	Yor
4	Reporting Data Analyst	37K–66K (Glassdoor est.)	ABOUT FANDUEL GROUP\n\nFanDuel Group is a worl...	3.9	FanDuel\n3.9	Yor

```
data_analyst_jobs['job_title']  =  data_analyst_jobs['job_title'].replace(['Sr.Dat
'Sr Data Analyst', 'sr data analyst','senior data analyst', 'Senior Data Analyst
'senior data analyst'],'Senior Data Analyst', regex=True)
data_analyst_jobs['job_title']  =  data_analyst_jobs['job_title'].replace(['Data  A
'Data Analyst Junior','data analyst junior','Junior Dat Analyst', 'Junior Data A
data_analyst_jobs['job_title']  =  data_analyst_jobs['job_title'].replace(['Data  A
'Middle Data Analyst'],'Middle Data Analyst', regex=True)
```

```
to_plot = data_analyst_jobs.job_title.value_counts()[:5]
color=sns.color_palette('Spectral')
to_plot
```

```
job_title
Data Analyst              405
Senior Data Analyst        99
Junior Data Analyst        58
Business Data Analyst      28
Sr. Data Analyst           21
Name: count, dtype: int64
```

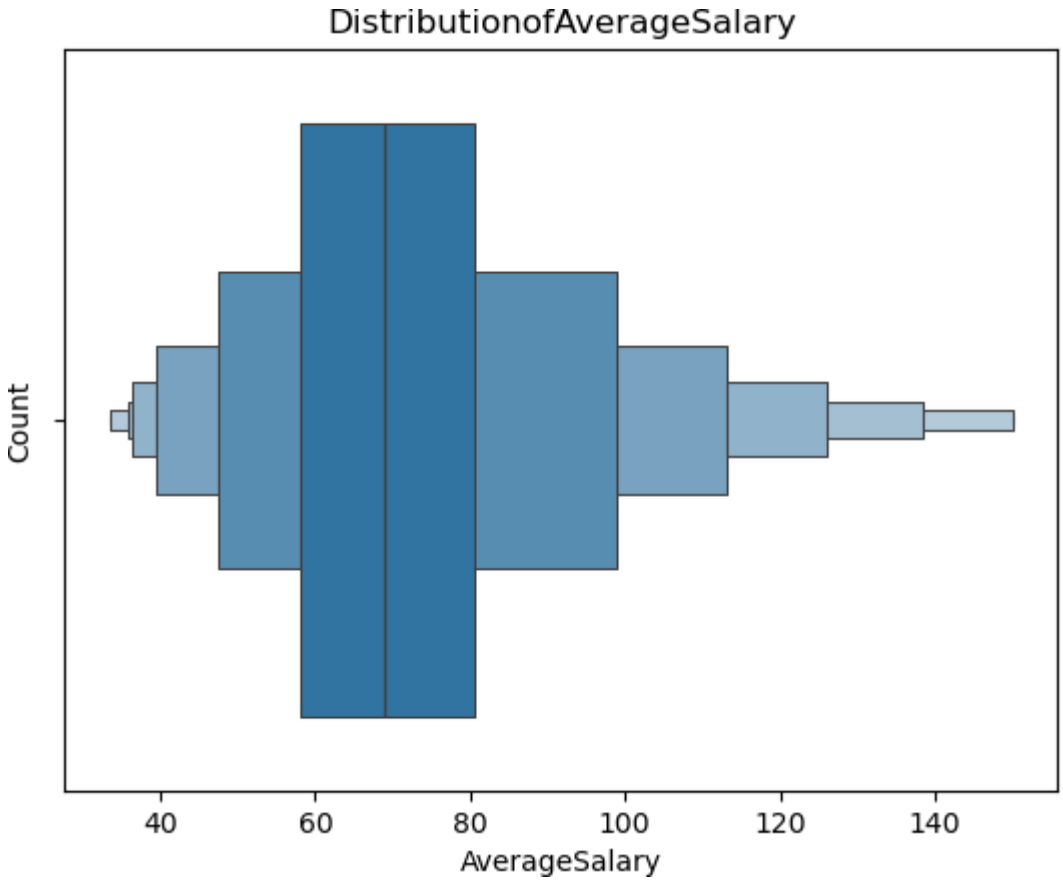
t

```
data_analyst_jobs[['MinSalary', 'MaxSalary']] = data_analyst_jobs['salary_estima
data_analyst_jobs['MinSalary'] = pd.to_numeric(data_analyst_jobs['MinSalary'])
data_analyst_jobs['MaxSalary'] = pd.to_numeric(data_analyst_jobs['MaxSalary'])
```

```
 data_analyst_jobs['MinSalary'] = data_analyst_jobs['MinSalary'].astype(float)
data_analyst_jobs['MaxSalary'] = data_analyst_jobs['MaxSalary'].astype(float)
data_analyst_jobs['average_salary'] = (data_analyst_jobs['MaxSalary'] + data_ana
data_analyst_jobs.drop(['salary_estimate', 'MinSalary','MaxSalary'], axis=1, inp
```

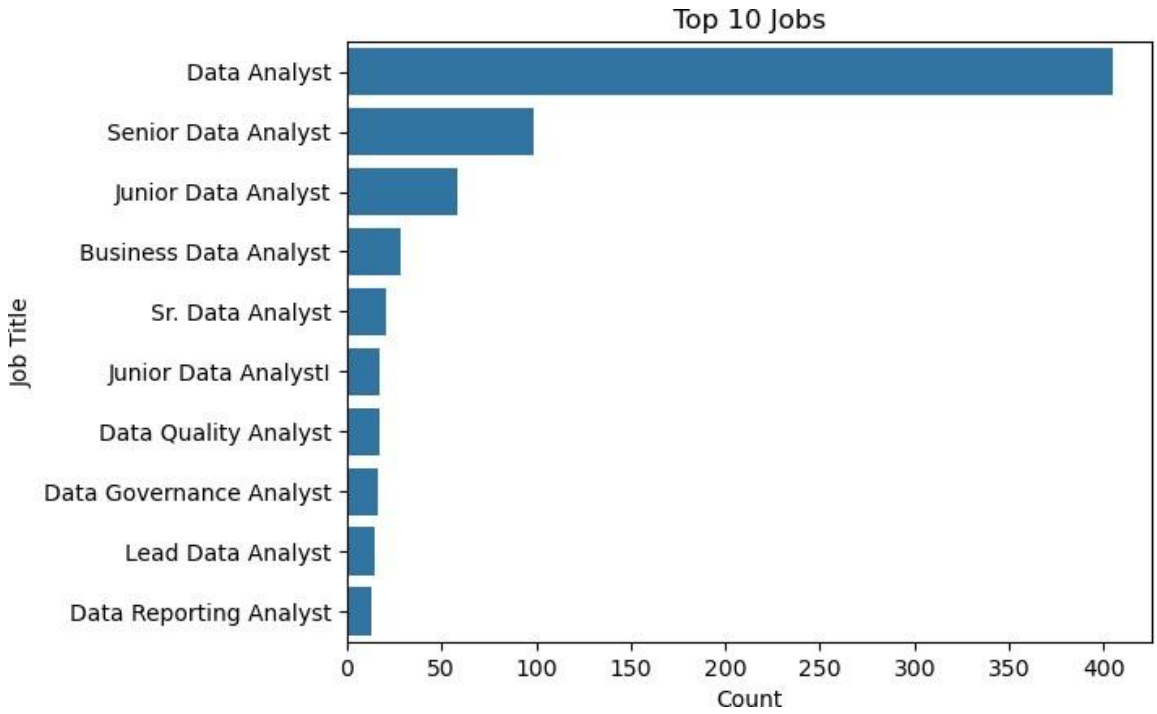
Average Salary

```
sns.boxenplot(data=data_analyst_jobs,x='average_salary')
plt.xlabel('AverageSalary')
plt.ylabel('Count')
plt.title('DistributionofAverageSalary')
plt.show()
```



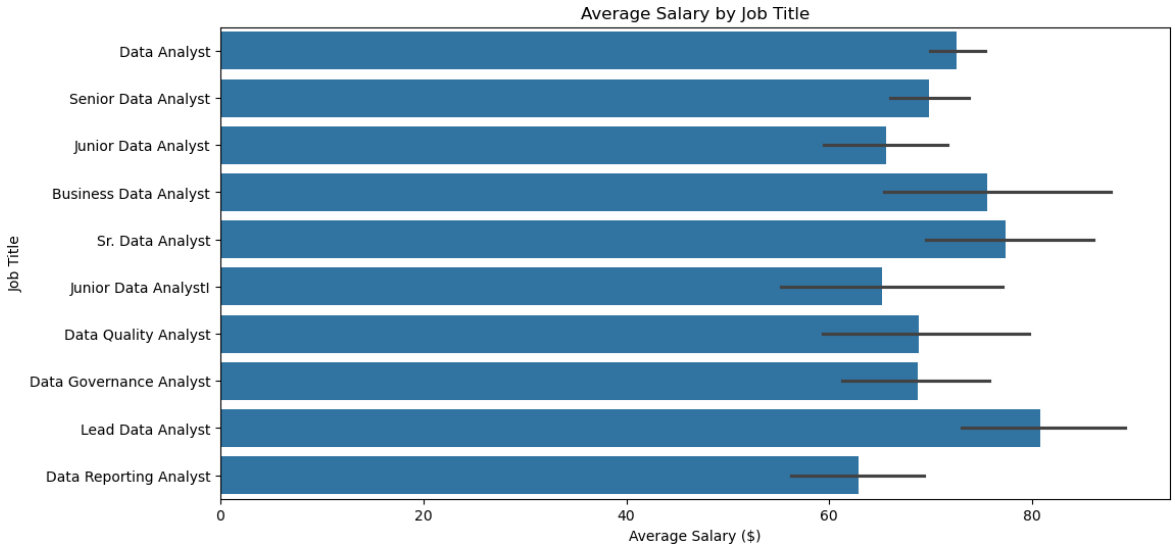

```
top_jobs = data_analyst_jobs['job_title'].value_counts().head(10)
sns.barplot(x=top_jobs.values, y=top_jobs.index)
plt.xlabel('Count')
plt.ylabel('Job Title')
plt.title('Top 10 Jobs')
plt.show()
```

t



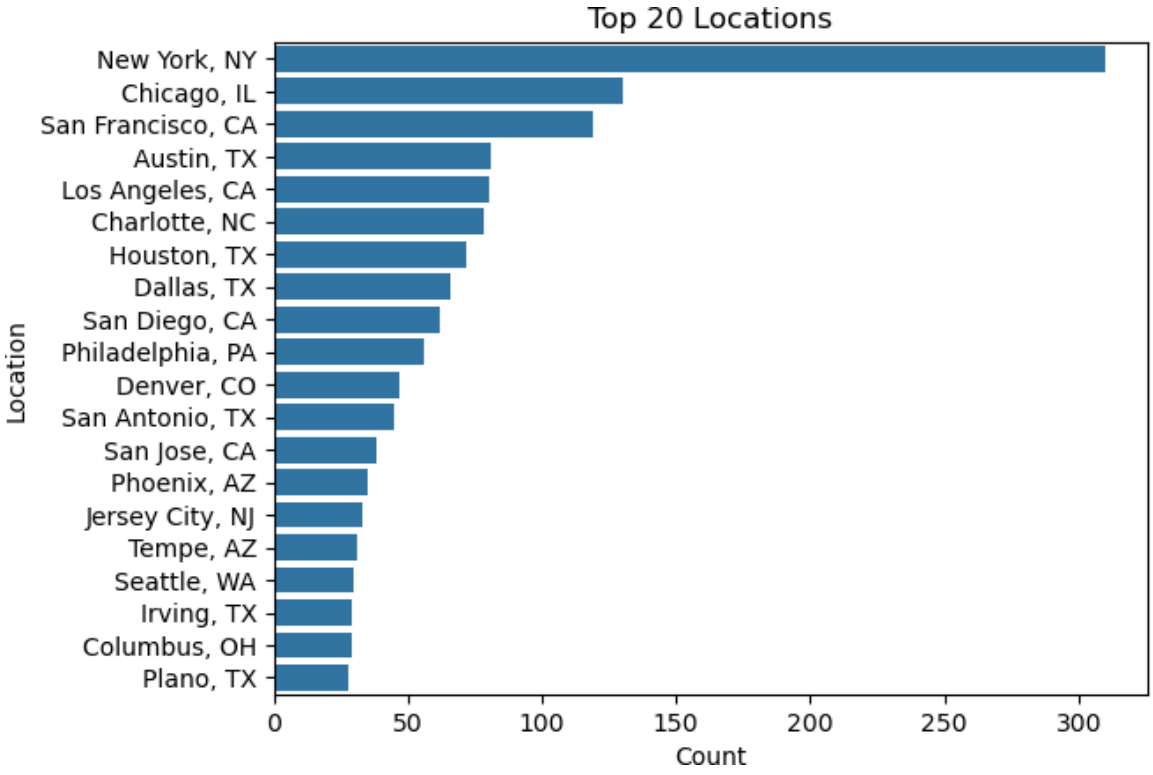
Salary and Job Title

```
data_analyst_jobs_sorted = data_analyst_jobs.sort_values(by='average_salary',asc
plt.figure(figsize=(12, 6))
sns.barplot(x='average_salary', y='job_title',
data=data_analyst_jobs_sorted, orient='h',
order=data_analyst_jobs_sorted['job_title'].value_counts().head(10).index)
plt.xlabel('Average Salary ($)')
plt.ylabel('Job Title')
plt.title('Average Salary by Job Title')
plt.show()
```

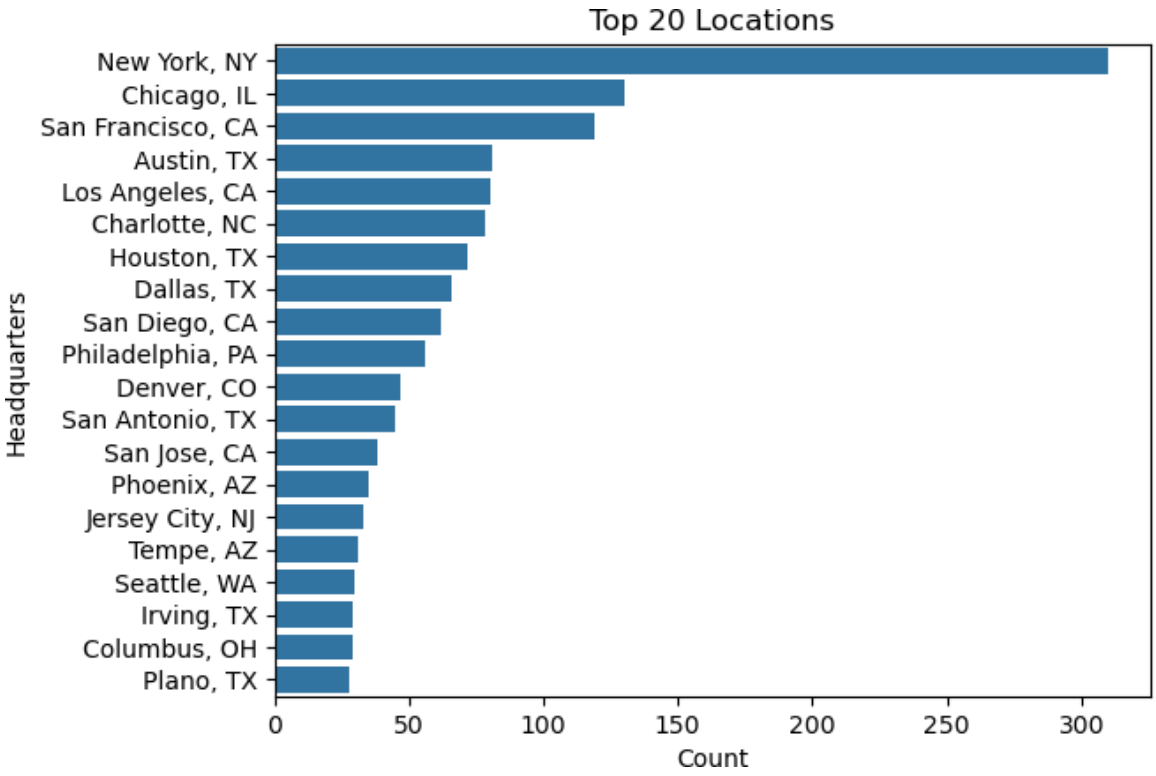



```
top_locations = data_analyst_jobs['location'].value_counts().head(20)
sns.barplot(x=top_locations.values, y=top_locations.index)
plt.xlabel('Count')
plt.ylabel('Location')
plt.title('Top 20 Locations')
plt.show()
```

t



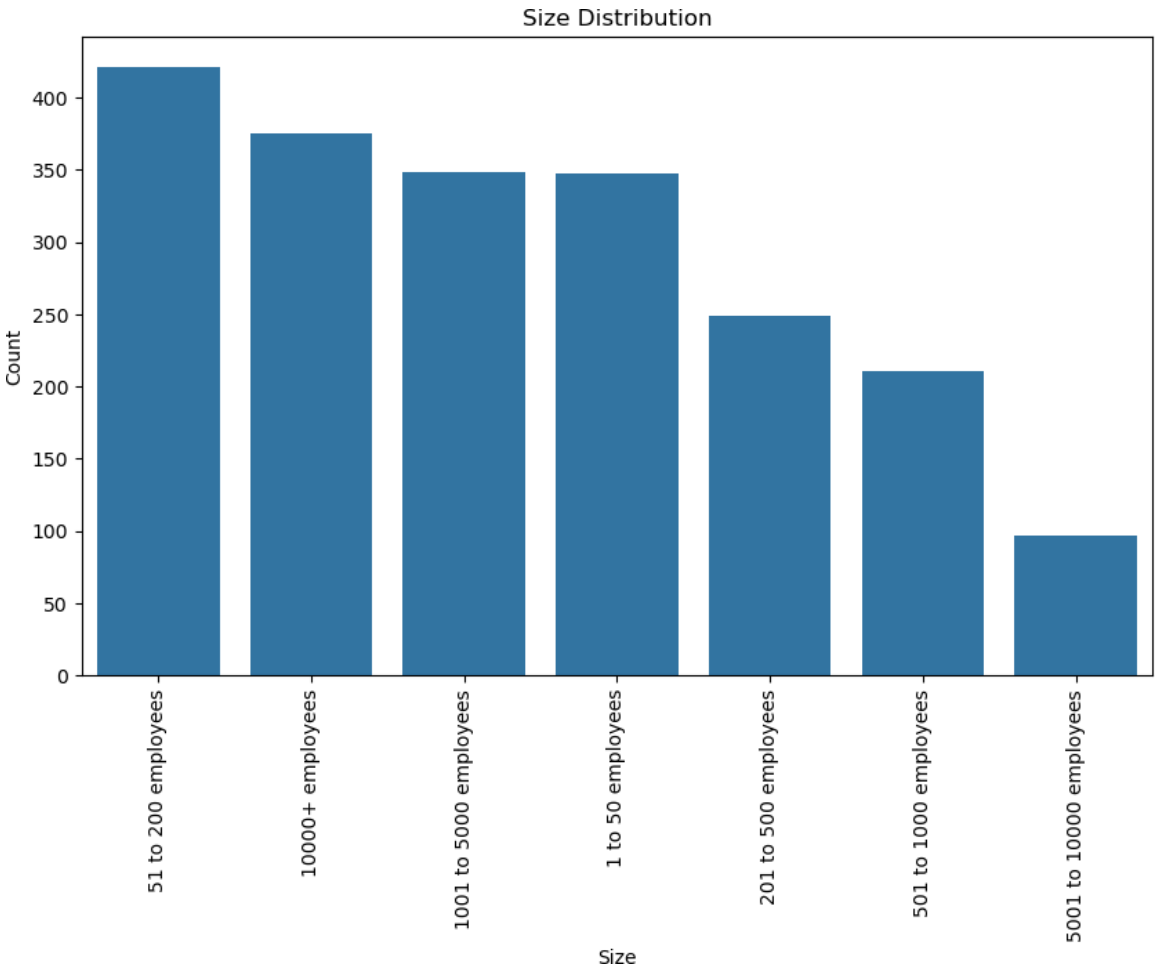

```
top_headquarters = data_analyst_jobs['headquarters'].value_counts().head(20)
sns.barplot(x=top_locations.values, y=top_locations.index)
plt.xlabel('Count')
plt.ylabel('Headquarters')
plt.title('Top 20 Locations')
plt.show()
```



Companies by Amount of Employee

t

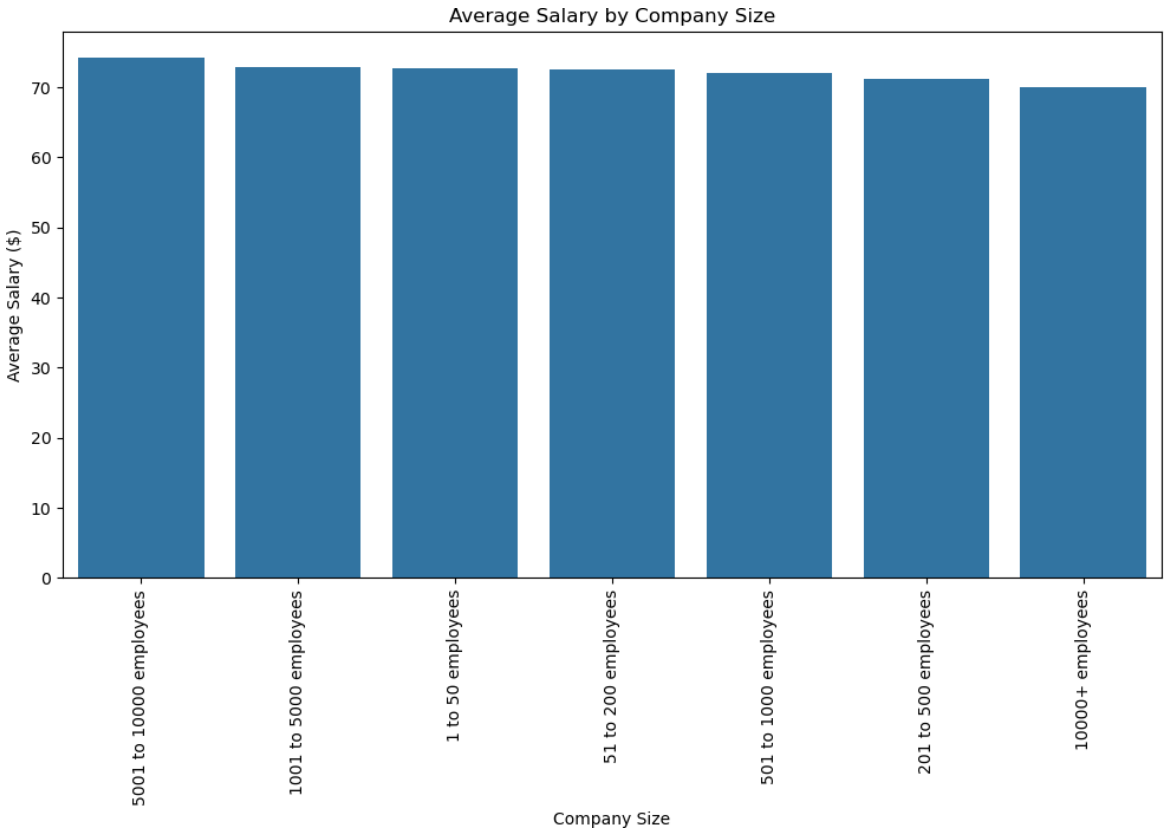
```
filtered_size = data_analyst_jobs[(data_analyst_jobs['size'] !='-1') & (data_ana
data_analyst_jobs_size = filtered_size['size'].value_counts().head(20)
plt.figure(figsize=(10,  6))
sns.barplot(x=data_analyst_jobs_size.index,y=data_analyst_jobs_size.values)
plt.xlabel('Size')
plt.ylabel('Count')
plt.title('Size Distribution')
plt.xticks(rotation=90)
plt.show()
```



Salary by Company Size

```
data_analyst_jobs_filtered  =  data_analyst_jobs[(data_analyst_jobs['size']  !=  '-1
data_analyst_jobs_sizeXsalary  =  data_analyst_jobs_filtered.groupby('size')['aver
data_analyst_jobs_sizeXsalary  =  data_analyst_jobs_sizeXsalary.sort_values(by='av
plt.figure(figsize=(12, 6))
sns.barplot(x='size', y='average_salary',data=data_analyst_jobs_sizeXsalary)
plt.xlabel('Company Size')
plt.ylabel('Average Salary ($)')
plt.title('Average Salary by Company Size')
plt.xticks(rotation=90)
plt.show()
```

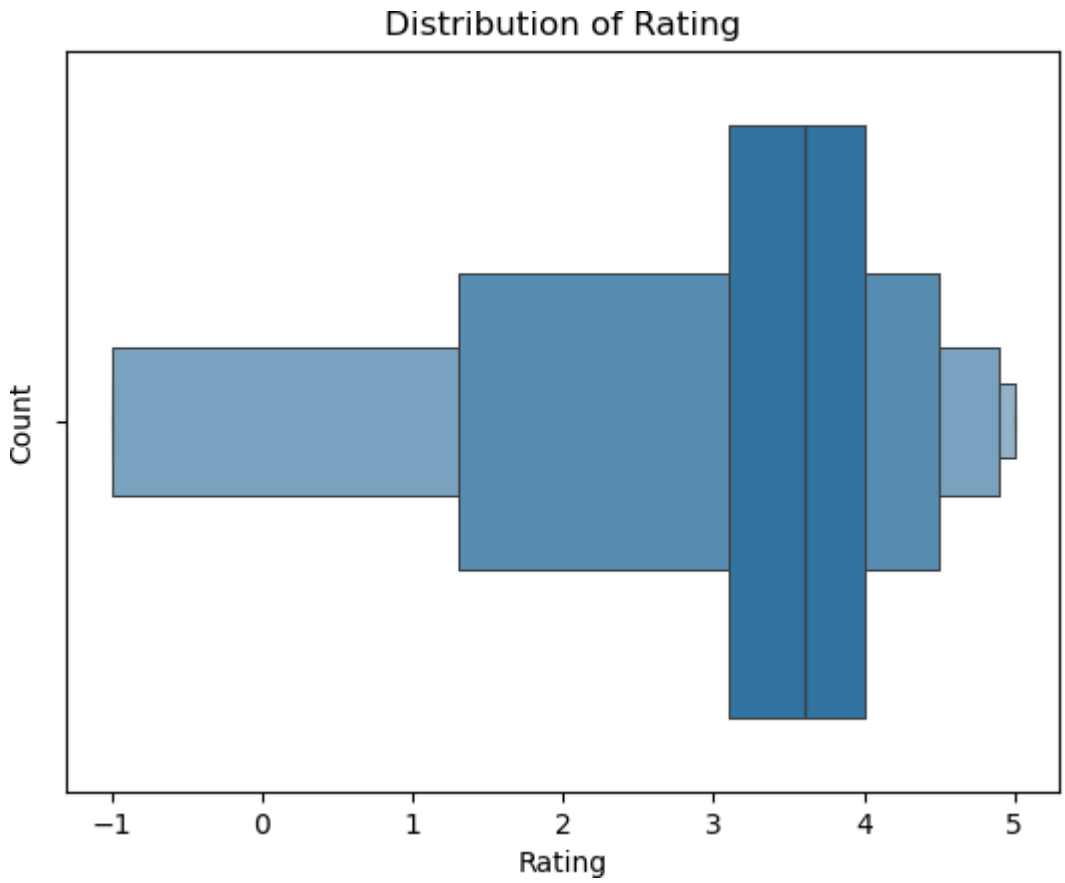
t



Company Rating

```
sns.boxenplot(data=data_analyst_jobs, x='Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Distribution of Rating')
plt.show()
```

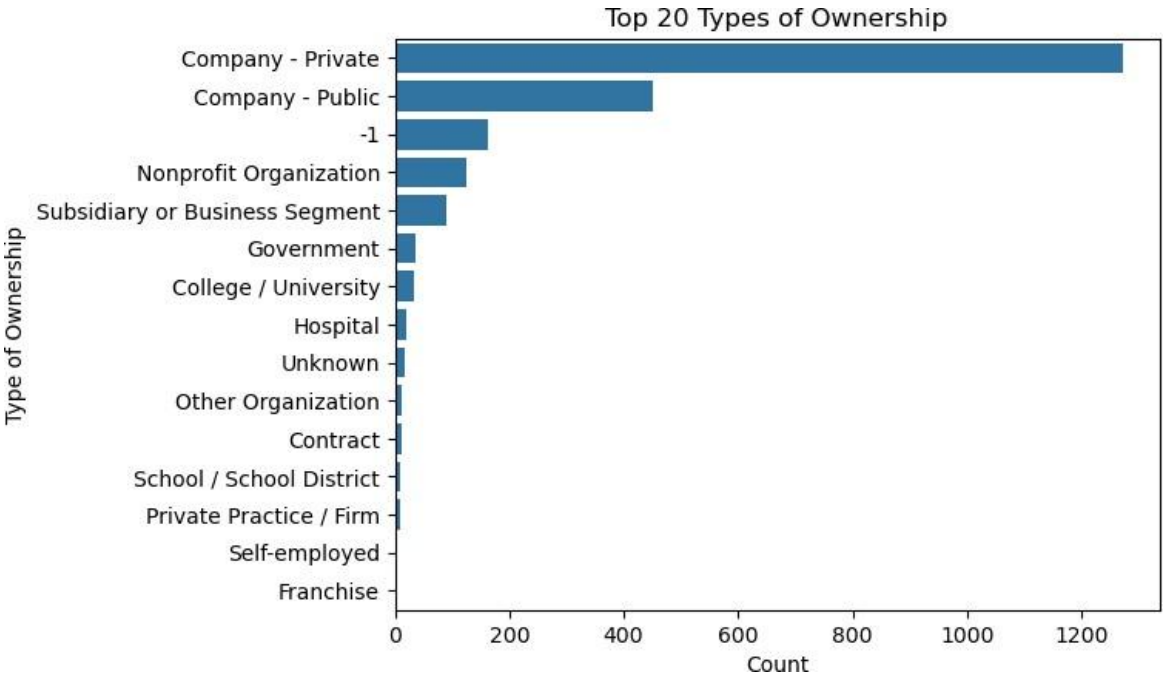
t



Type of Ownership

```
TOP = data_analyst_jobs[(data_analyst_jobs['type_of_ownership']
!= '-1') & (data_analyst_jobs['type_of_ownership'] !='Unknown')]
TOP = data_analyst_jobs['type_of_ownership'].value_counts().head(20)
sns.barplot(x=TOP.values, y=TOP.index)
plt.xlabel('Count')
plt.ylabel('Type of Ownership')
plt.title('Top 20 Types of Ownership')
plt.show()
```

t



Top Sectors

In [284…

```
data_analyst_jobs_sector = data_analyst_jobs[data_analyst_jobs['sector'] !='-1']
sns.barplot(x=data_analyst_jobs_sector.values,y=data_analyst_jobs_sector.index)
plt.xlabel('Count')
plt.ylabel('Sector')
plt.title('Sector Distribution')
plt.show()
```