

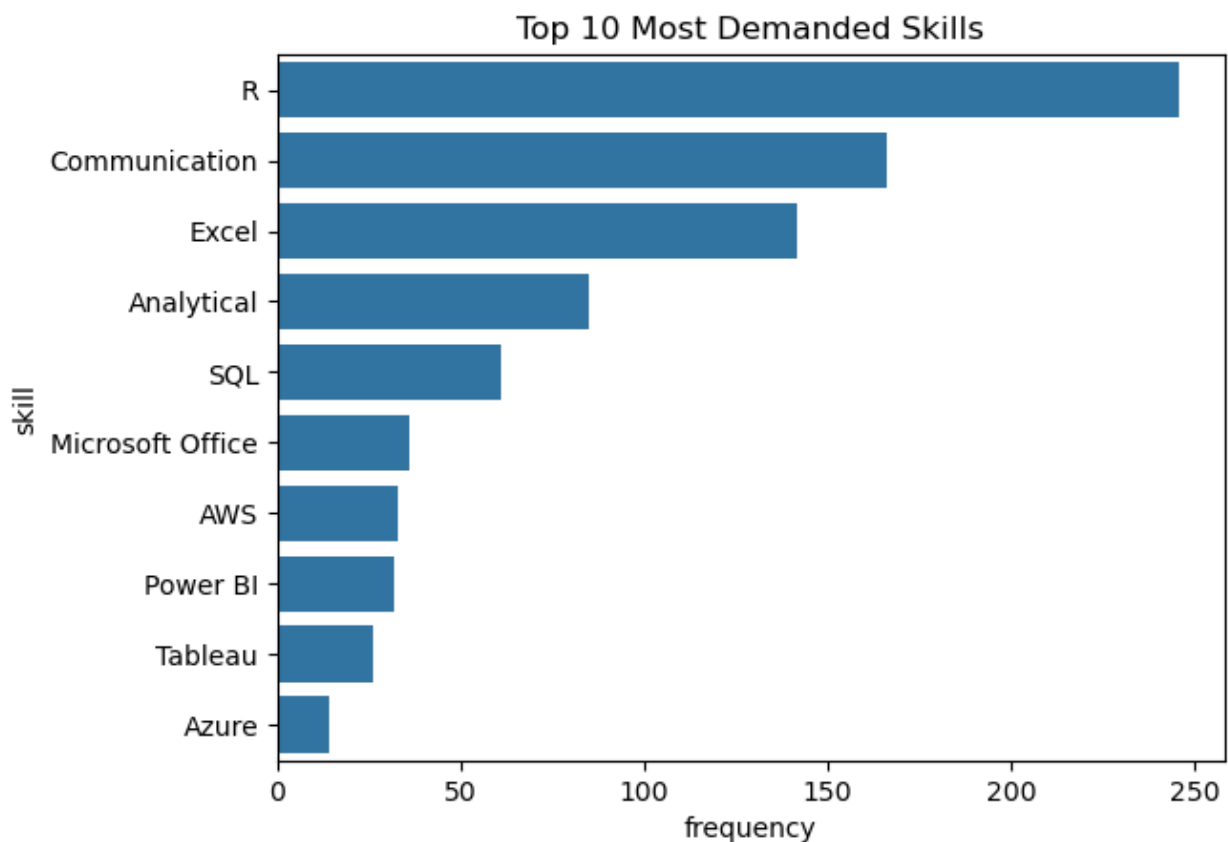
IMPORTING IMPORTANT LIBRARIES

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

WILL READ EACH QUERY RESULT CSV INDIVIDUALLY

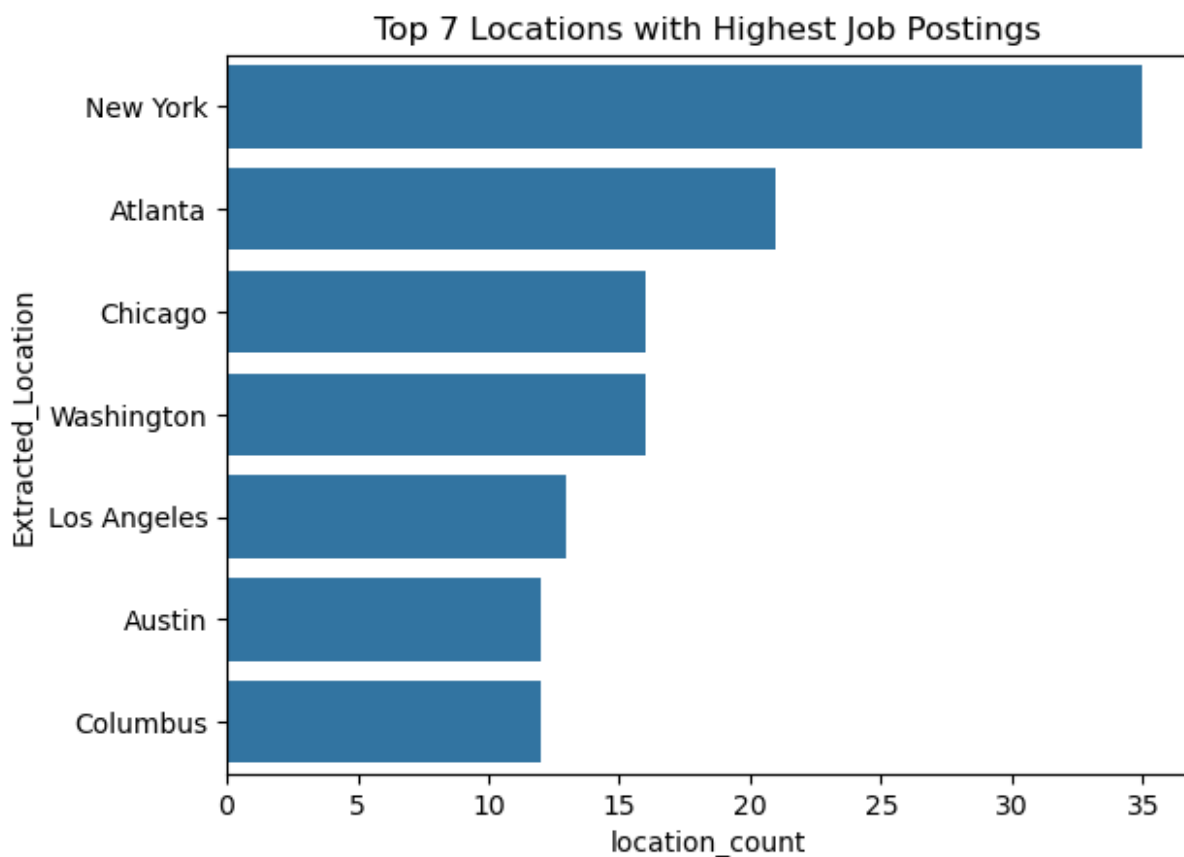
```
import pandas as pd
topskills = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR VISUALISATION/top10skills.csv')

sns.barplot(data=topskills, x="frequency", y="skill")
plt.title("Top 10 Most Demanded Skills")
plt.show()
```



```
BestLocation = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR VISUALISATION/Top7locations.csv')
```

```
sns.barplot(data=BestLocation, x="location_count",  
y='Extracted_Location', estimator=sum)  
plt.title("Top 7 Locations with Highest Job Postings")  
plt.show()
```

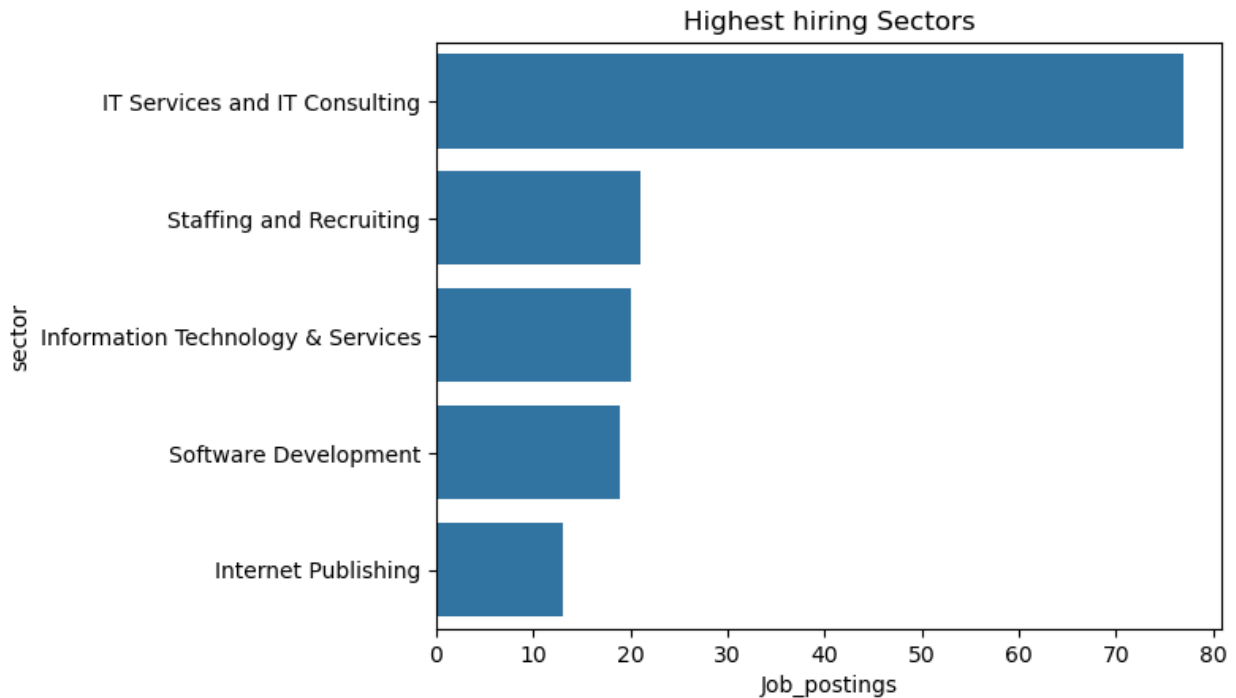


```
TopSectors = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR VISUALISATION/TopSectors.csv')
```

```
TopSectors
```

	sector	Job_postings
0	IT Services and IT Consulting	77
1	Staffing and Recruiting	21
2	Information Technology & Services	20
3	Software Development	19
4	Internet Publishing	13

```
sns.barplot(data=TopSectors, x="Job_postings", y='sector')  
plt.title("Highest hiring Sectors")  
plt.show()
```



```
WorstSectors = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR VISUALISATION/Least_Posting_Sectors.csv')
```

WORST SECTORS THAT HAVE THE LEAST JOB POSTINGS FOR BUSINESS ANALYSTS

WorstSectors

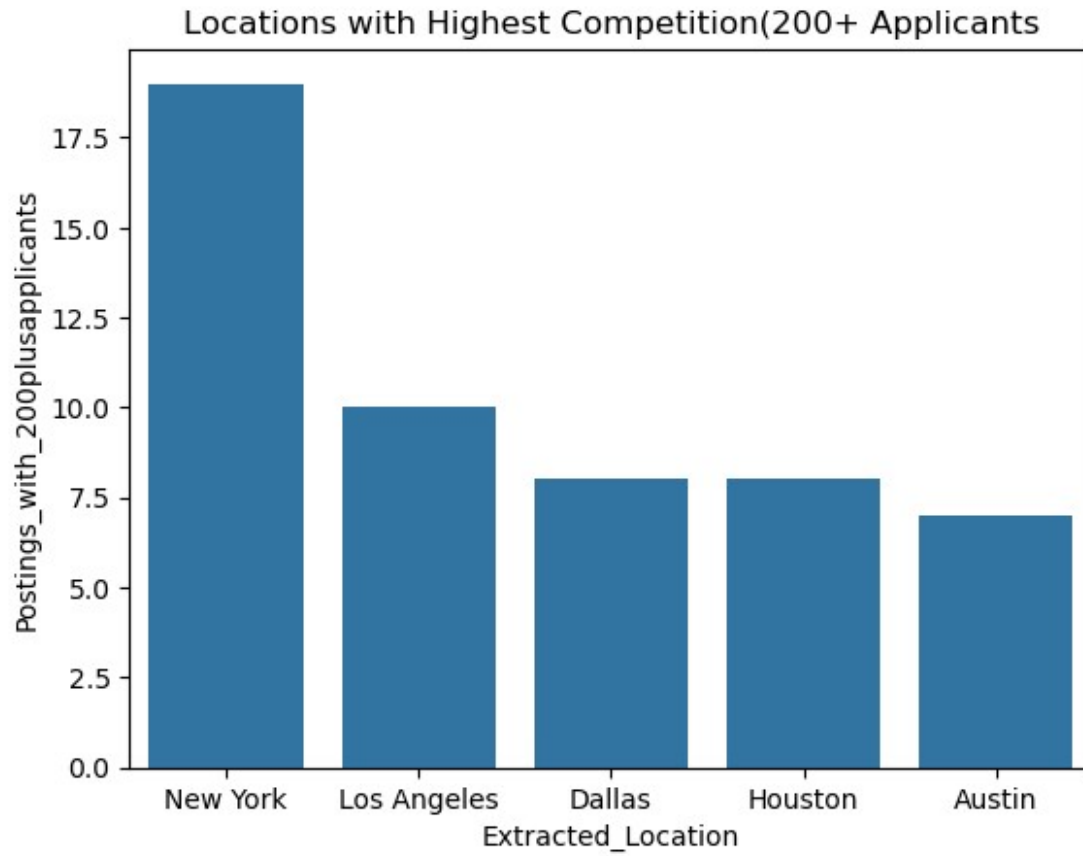
	sector	Postings
0	Investment Banking, Investment Management, and...	1
1	Accounting, Legal Services, and Law Practice	1
2	Motor Vehicle Manufacturing	1
3	Environmental Services	1
4	Wholesale Furniture and Home Furnishings	1
5	Staffing and Recruiting, Retail, and Personal ...	1
6	Utilities, Construction, and Oil and Gas	1
7	Retail and Business Consulting and Services	1
8	IT Services and IT Consulting and Airlines and...	1
9	Investment Banking	1
10	Public Relations and Communications Services	1
11	Hospitals and Health Care, Hospitals, and Medi...	1
12	Health and Human Services	1

13	Law Practice	1
14	Airlines and Aviation	1

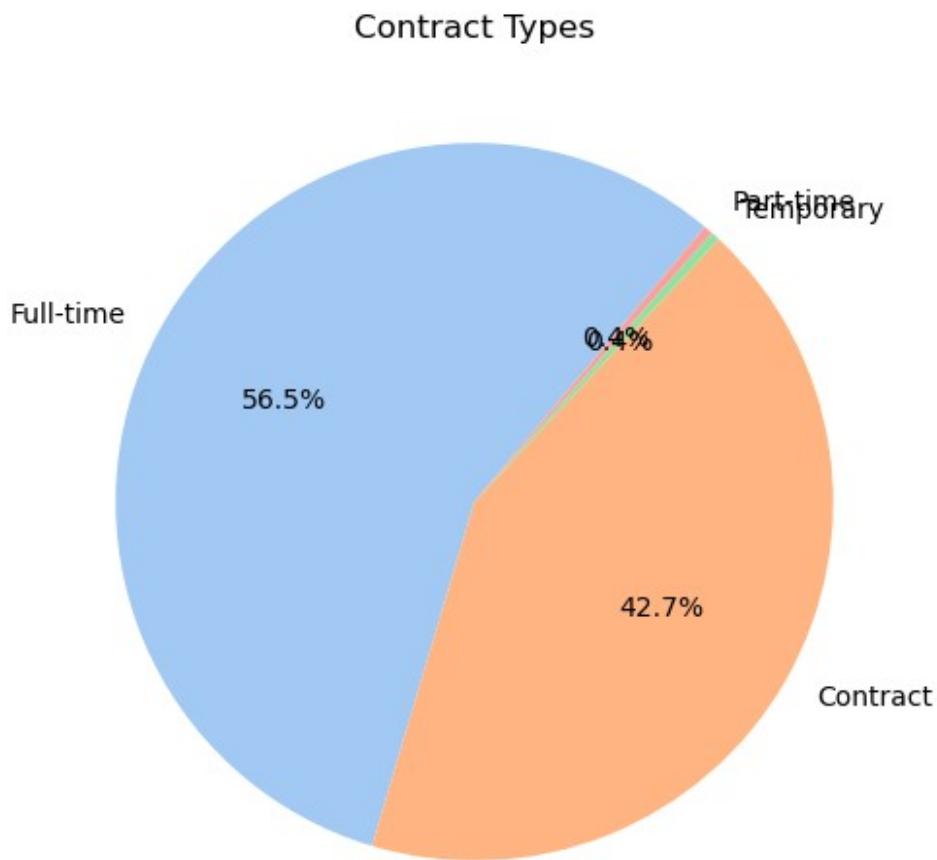
```
CompetetiveLocations =
pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR
VISUALISATION/Highestapplicantlocations.csv')
CompetetiveLocations
```

	Extracted_Location	Postings_with_200plusapplicants
0	New York	19
1	Los Angeles	10
2	Dallas	8
3	Houston	8
4	Austin	7
5	Raleigh	6
6	Atlanta	5
7	Chicago	5
8	Washington	5
9	Philadelphia	4
10	Charlotte	3
11	Minneapolis	3
12	Columbus	3
13	Boston	3
14	Tampa	2
15	San Diego	2
16	Seattle	2
17	Nashville	2
18	Kansas City	2
19	San Antonio	2
20	Baltimore	1
21	Orlando	1
22	Denver	1
23	Miami	1

```
sns.barplot(data=CompetetiveLocations.head(5), x="Extracted_Location",
y="Postings_with_200plusapplicants")
plt.title("Locations with Highest Competition(200+ Applicants)")
plt.show()
```



```
Contract = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR  
VISUALISATION/Contractype.csv')  
colors = sns.color_palette('pastel')  
  
plt.figure(figsize=(6,6))  
plt.pie(Contract['TotalNUM'], labels=Contract.iloc[:,0],  
autopct='%1.1f%%', startangle=50, colors=colors)  
plt.title('Contract Types')  
plt.show()
```

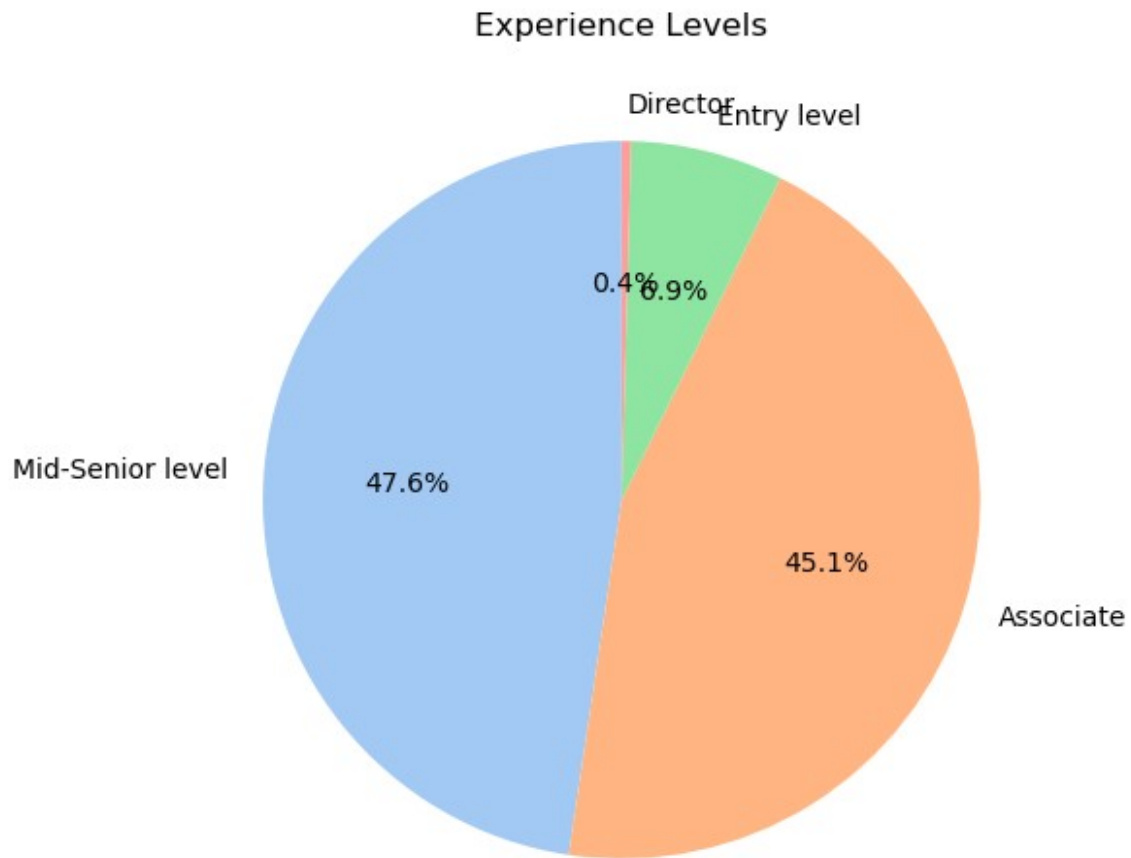


Experience

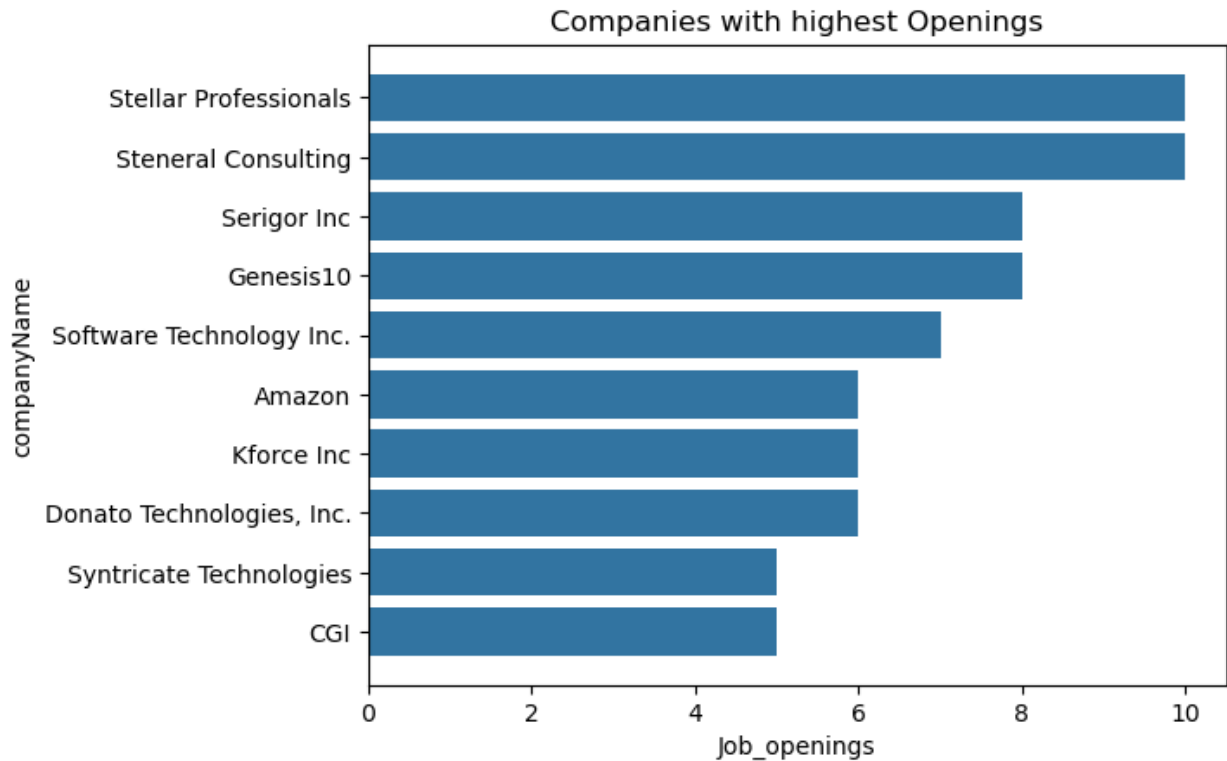
	experienceLevel	role_count
0	Mid-Senior level	111
1	Associate	105
2	Entry level	16
3	Director	1

```
Experience = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR VISUALISATION/Experiencelevel.csv')
colors = sns.color_palette('pastel')
```

```
plt.figure(figsize=(6,6))
plt.pie(Experience['role_count'], labels=Experience.iloc[:,0],
autopct='%1.1f%%', startangle=90, colors=colors)
plt.title('Experience Levels')
plt.show()
```



```
TopCompanies = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR  
VISUALISATION/TopCompanies.csv')  
  
sns.barplot(data=TopCompanies, x='Job_openings', y='companyName')  
plt.title('Companies with highest Openings')  
plt.plot()  
[]
```



```
Timeline = pd.read_csv(r'/Users/divjyotsinghsuri/Desktop/FOR  
VISUALISATION/PostingsTimeline.csv')  
  
Timeline['date'] = pd.to_datetime(Timeline[['year',  
'month']].assign(day=1))  
  
sns.barplot(x='date', y='job_count', data=Timeline, color='skyblue')  
plt.title('Job Postings Over Time')  
plt.xlabel('Date')  
plt.ylabel('Job Count')  
plt.xticks(rotation=45)  
plt.show()
```