Data Analyst Project

Job Market Analysis

Team Members : Karunakaran R

Email id : rkarunakaranraja@gmail.com

Batch : 12th May CDA

Project id : PTID-CDA-SEP-25-722

Time period : 24th Sep to 25th Sep

Introduction:

This project is to analyse the job market trends for positions by analysing job data. It identifies states with the highest job opportunities, compares salary ranges, highlights top industries and companies and evaluates skills in demand and provide insights that can inform job seekers and employers.

Objectives:

- Identify states with most job openings.
- Compare minimum and maximum salaries across states.
- Find top industries and companies hiring.
- Analysis top job titles and required skills.
- Examine education vs salary trends

Dataset details:

My dataset is includes 742 rows and 42 features like Job title, Salary Estimate, Job Description, Rating, Company, Location, Company, Headquarters and many more acquired from various sources.

Tools & methodology:

Python:

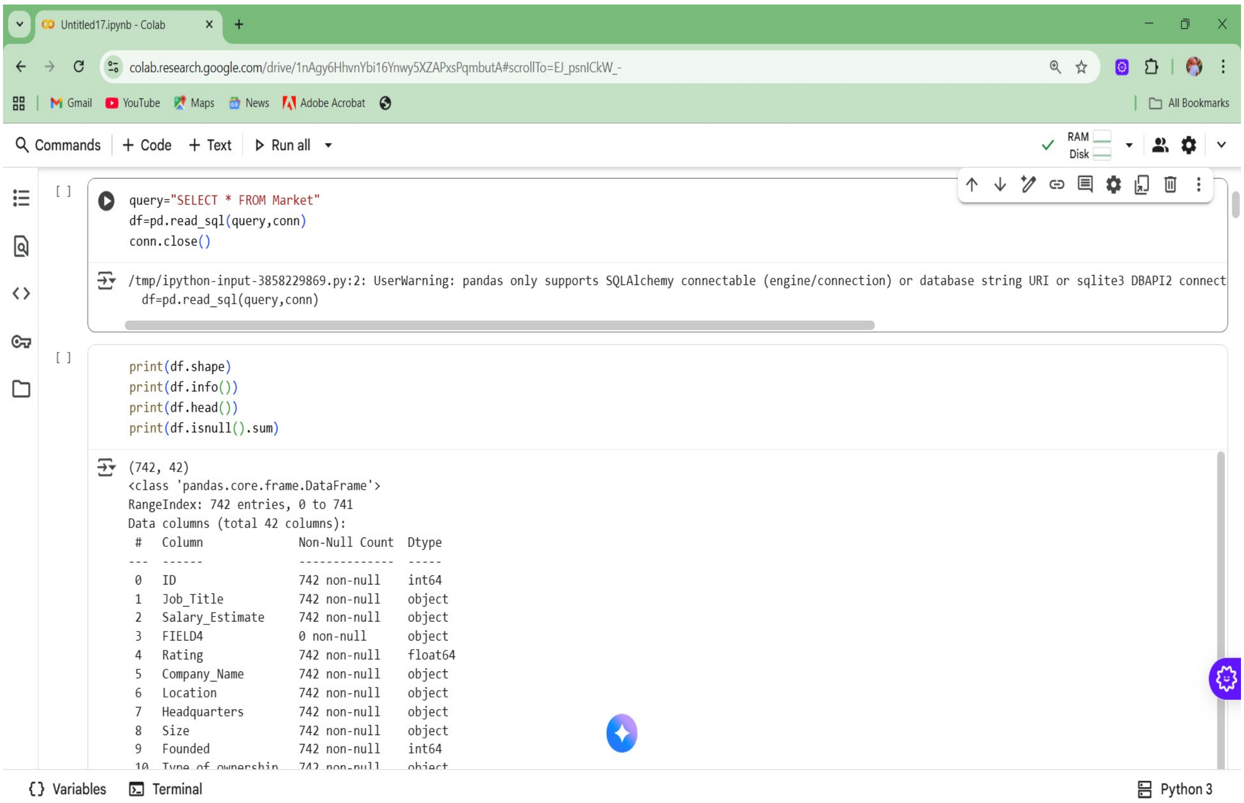
- Extract data
- Clean and prepare dataset
- Load into Tableau

Tableau:

- Build dashboards (with multiple charts)

Python results:

I first import libraries like pandas, numpy, matplotlib. And I connect the database into python using given database crendential.



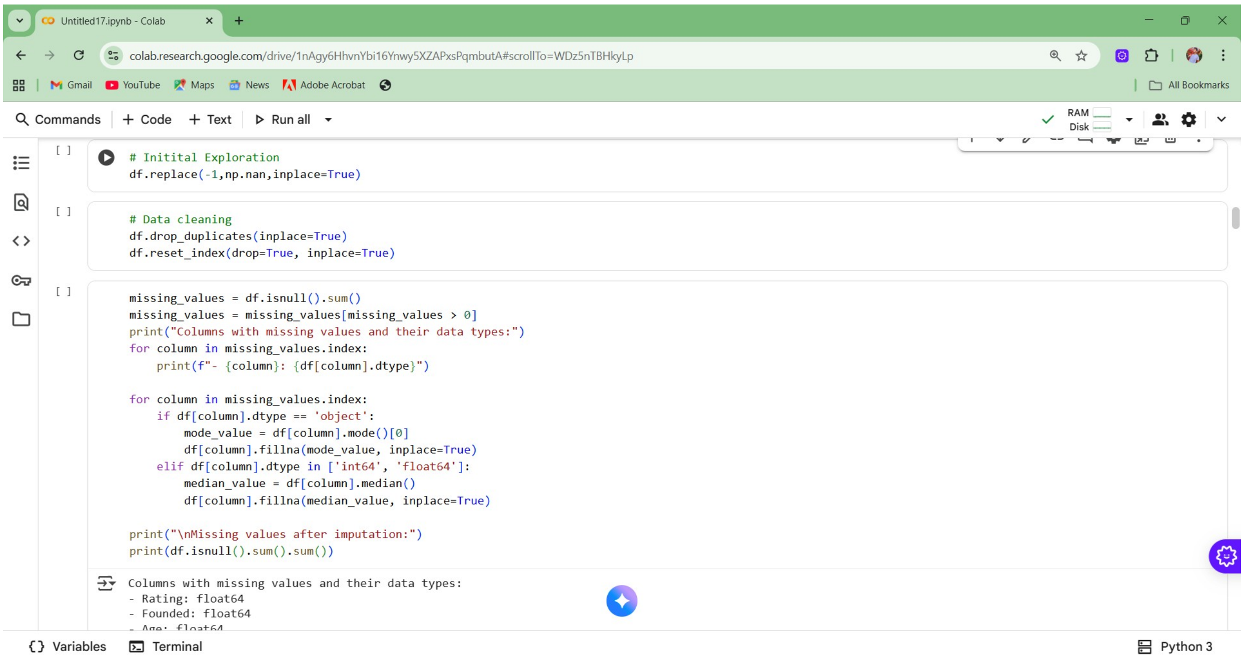
```
query="SELECT * FROM Market"
df=pd.read_sql(query,conn)
conn.close()
```

```
/tmp/ipython-input-3858229869.py:2: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connect
  df=pd.read_sql(query,conn)
```

```
print(df.shape)
print(df.info())
print(df.head())
print(df.isnull().sum)
```

```
(742, 42)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 742 entries, 0 to 741
Data columns (total 42 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  742 non-null    int64
 1   Job_Title           742 non-null    object
 2   Salary_Estimate     742 non-null    object
 3   FIELD4              0 non-null      object
 4   Rating              742 non-null    float64
 5   Company_Name        742 non-null    object
 6   Location            742 non-null    object
 7   Headquarters        742 non-null    object
 8   Size                742 non-null    object
 9   Founded             742 non-null    int64
 10  Type_of_ownership   742 non-null    object
```

I started EDA process using python like data cleaning, removing duplicate, analysis the data and handling null values and data format changing which be a cleaned data for doing further process to checking like states with most number of Jobs, average minimal and maximal salaries in different states, average salary in different states, top 5 industries with maximum number of data science related job postings, companies with maximum number of job openings, job titles with most number of jobs, salary of job titles with most number of jobs, skills required by companies for each job title, relation between average salary and education.



```
# Initital Exploration
df.replace(-1,np.nan,inplace=True)
```

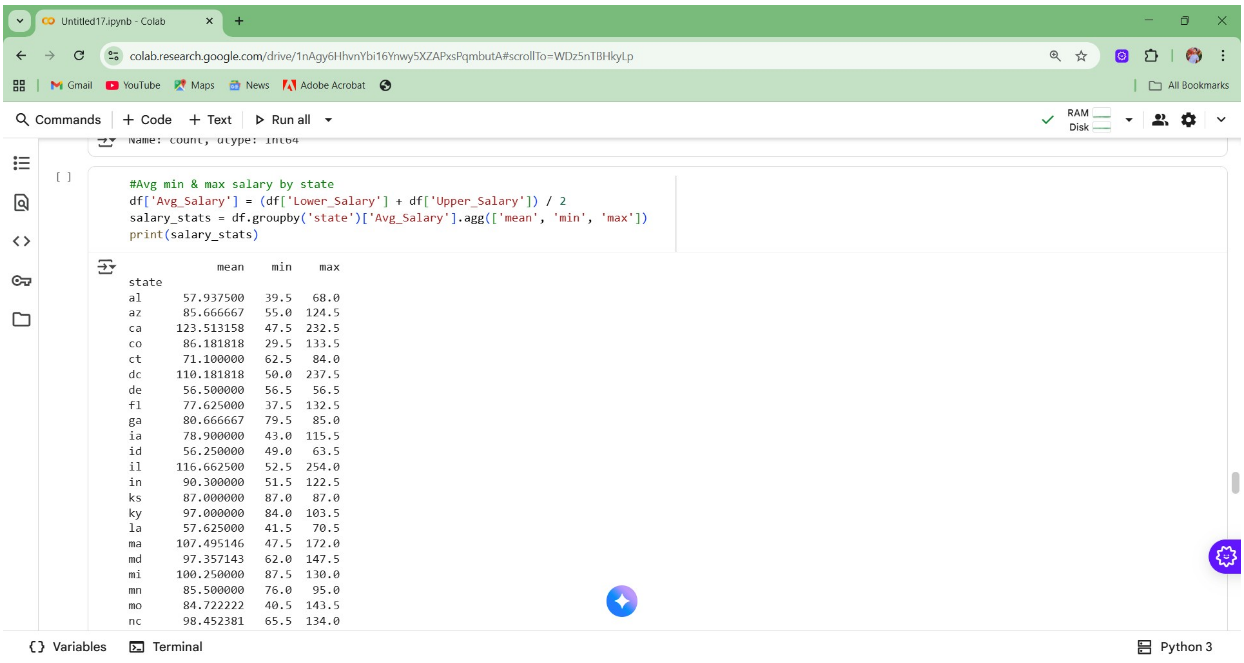
```
# Data cleaning
df.drop_duplicates(inplace=True)
df.reset_index(drop=True, inplace=True)
```

```
missing_values = df.isnull().sum()
missing_values = missing_values[missing_values > 0]
print("Columns with missing values and their data types:")
for column in missing_values.index:
    print(f"- {column}: {df[column].dtype}")

for column in missing_values.index:
    if df[column].dtype == 'object':
        mode_value = df[column].mode()[0]
        df[column].fillna(mode_value, inplace=True)
    elif df[column].dtype in ['int64', 'float64']:
        median_value = df[column].median()
        df[column].fillna(median_value, inplace=True)

print("\nMissing values after imputation:")
print(df.isnull().sum())
```

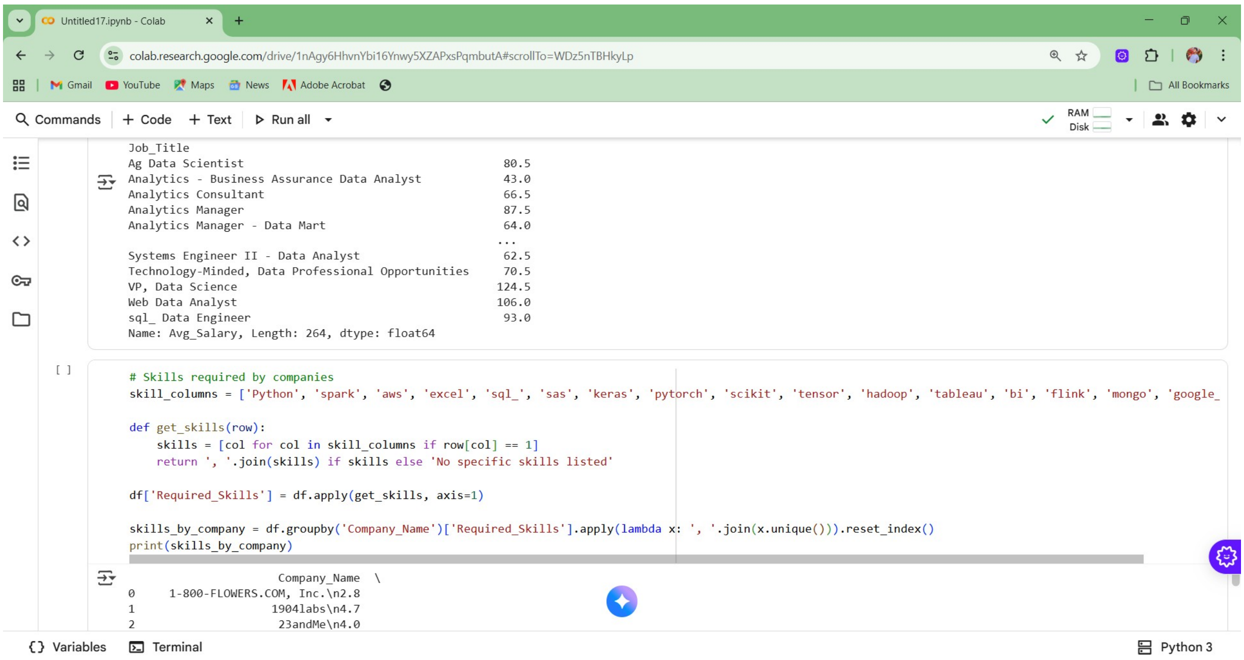
```
Columns with missing values and their data types:
- Rating: float64
- Founded: float64
- Age: float64
```



```
Name: count, dtype: int64
```

```
#Avg min & max salary by state
df['Avg_Salary'] = (df['Lower_Salary'] + df['Upper_Salary']) / 2
salary_stats = df.groupby('state')['Avg_Salary'].agg(['mean', 'min', 'max'])
print(salary_stats)
```

```
             mean   min    max
state
al      57.937500  39.5   68.0
az      85.666667  55.0  124.5
ca     123.513158  47.5  232.5
co      86.181818  29.5  133.5
ct      71.100000  62.5   84.0
dc     110.181818  50.0  237.5
de      56.500000  56.5   56.5
fl      77.625000  37.5  132.5
ga      80.666667  79.5   85.0
ia      78.900000  43.0  115.5
id      56.250000  49.0   63.5
il     116.662500  52.5  254.0
in      90.300000  51.5  122.5
ks      87.000000  87.0   87.0
ky      97.000000  84.0  103.5
la      57.625000  41.5   70.5
ma     107.495146  47.5  172.0
md      97.357143  62.0  147.5
mi     100.250000  87.5  130.0
mn      85.500000  76.0   95.0
mo      84.722222  40.5  143.5
nc      98.452381  65.5  134.0
```



```
Job_Title
Ag Data Scientist                                      80.5
Analytics - Business Assurance Data Analyst            43.0
Analytics Consultant                                   66.5
Analytics Manager                                      87.5
Analytics Manager - Data Mart                          64.0
                                                       ...
Systems Engineer II - Data Analyst                     62.5
Technology-Minded, Data Professional Opportunities     70.5
VP, Data Science                                      124.5
Web Data Analyst                                      106.0
sql_ Data Engineer                                     93.0
Name: Avg_Salary, Length: 264, dtype: float64
```

[ ]

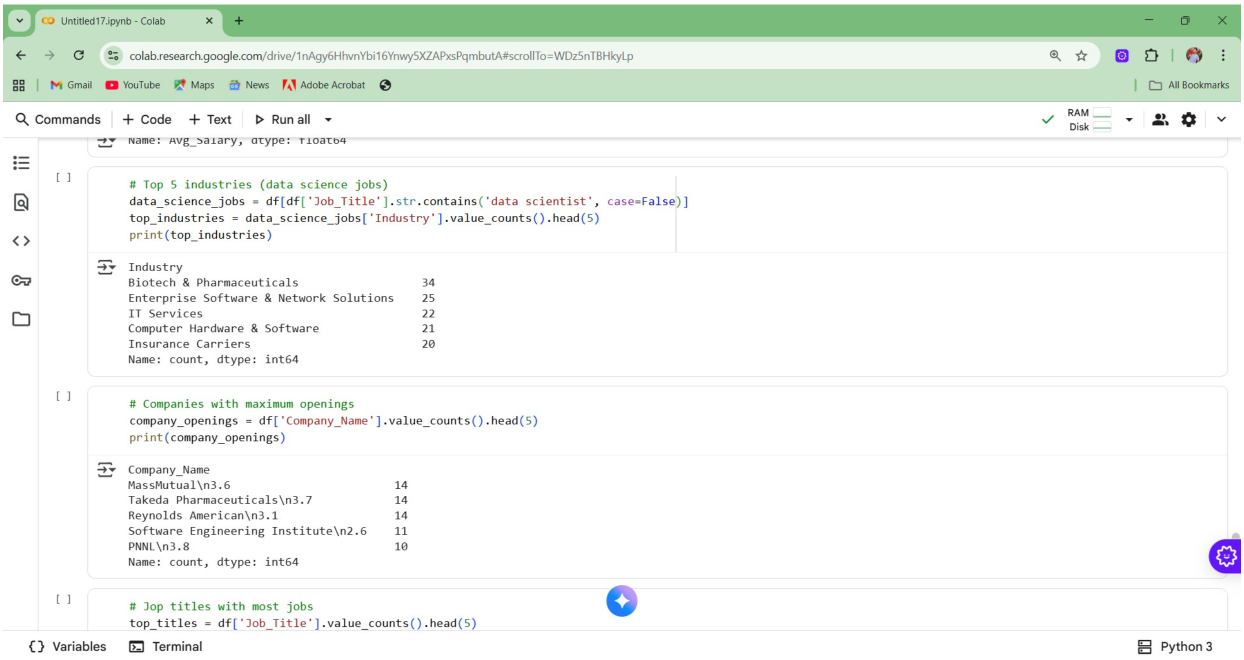
```
# Skills required by companies
skill_columns = ['Python', 'spark', 'aws', 'excel', 'sql_', 'sas', 'keras', 'pytorch', 'scikit', 'tensor', 'hadoop', 'tableau', 'bi', 'flink', 'mongo', 'google_

def get_skills(row):
    skills = [col for col in skill_columns if row[col] == 1]
    return ', '.join(skills) if skills else 'No specific skills listed'

df['Required_Skills'] = df.apply(get_skills, axis=1)

skills_by_company = df.groupby('Company_Name')['Required_Skills'].apply(lambda x: ', '.join(x.unique())).reset_index()
print(skills_by_company)
```

```
                    Company_Name  \
0     1-800-FLOWERS.COM, Inc.\n2.8
1                   1904labs\n4.7
2                    23andMe\n4.0
```



```
Name: Avg_Salary, dtype: float64
```

[ ]

```
# Top 5 industries (data science jobs)
data_science_jobs = df[df['Job_Title'].str.contains('data scientist', case=False)]
top_industries = data_science_jobs['Industry'].value_counts().head(5)
print(top_industries)
```

```
Industry
Biotech & Pharmaceuticals                   34
Enterprise Software & Network Solutions     25
IT Services                                 22
Computer Hardware & Software                21
Insurance Carriers                          20
Name: count, dtype: int64
```

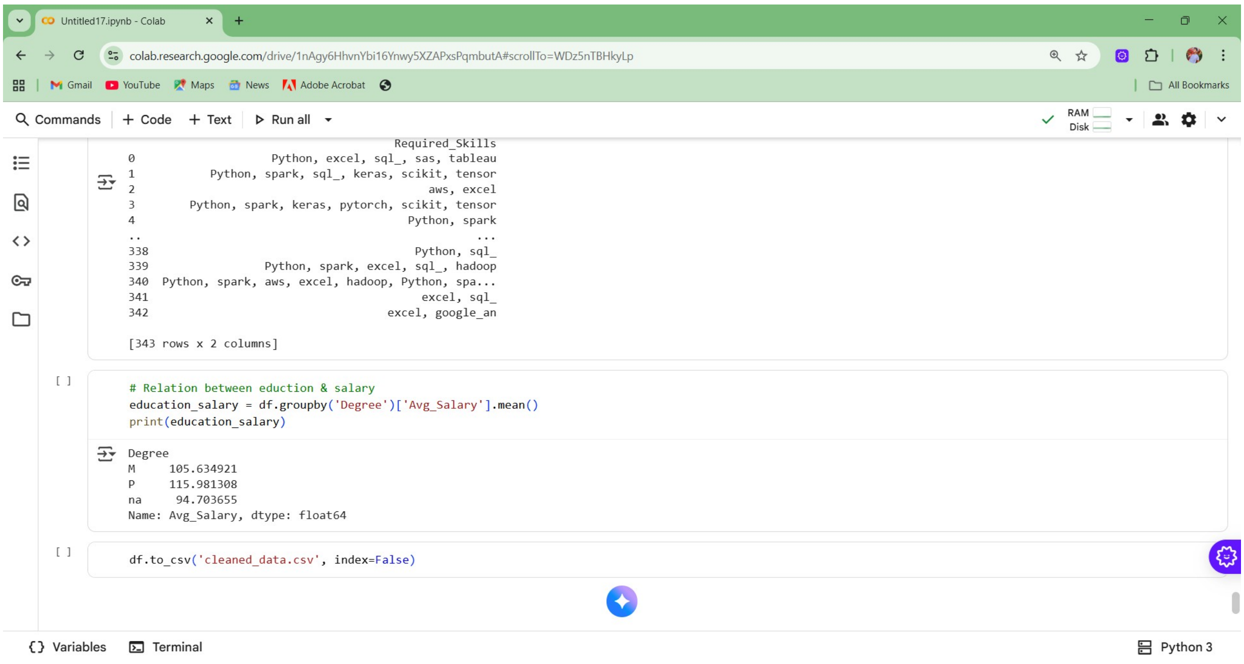
[ ]

```
# Companies with maximum openings
company_openings = df['Company_Name'].value_counts().head(5)
print(company_openings)
```

```
Company_Name
MassMutual\n3.6                         14
Takeda Pharmaceuticals\n3.7             14
Reynolds American\n3.1                  14
Software Engineering Institute\n2.6     11
PNNL\n3.8                               10
Name: count, dtype: int64
```

[ ]

```
# Jop titles with most jobs
top_titles = df['Job_Title'].value_counts().head(5)
```



```
                                      Required_Skills
0                 Python, excel, sql_, sas, tableau
1            Python, spark, sql_, keras, scikit, tensor
2                                          aws, excel
3          Python, spark, keras, pytorch, scikit, tensor
4                                       Python, spark
..                                                ...
338                                       Python, sql_
339             Python, spark, excel, sql_, hadoop
340  Python, spark, aws, excel, hadoop, Python, spa...
341                                       excel, sql_
342                                 excel, google_an

[343 rows x 2 columns]
```

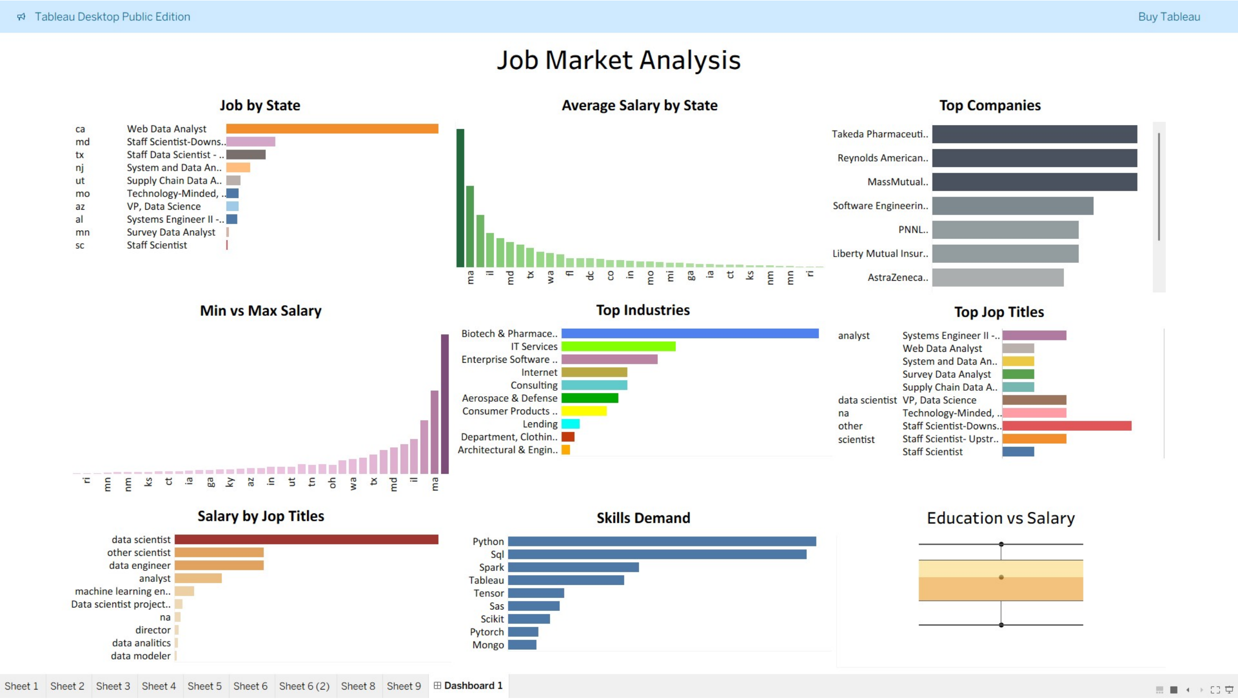
```
# Relation between eduction & salary
education_salary = df.groupby('Degree')['Avg_Salary'].mean()
print(education_salary)
```

```
Degree
M     105.634921
P     115.981308
na     94.703655
Name: Avg_Salary, dtype: float64
```

```
df.to_csv('cleaned_data.csv', index=False)
```

Tableau Dashboard:

Finally enter into tableau. I get the csv cleaned file which I downloaded from python. I creates dashboard with using charts which will be easy to understand for business peoples and then with using the help of this dashboards they can make decisions.



- Jobs by State → Which state has more jobs
- Min vs Max Salary → Salary distribution across states
- Top Industries → Industries hiring the most
- Top Companies → Who recruits more
- Top Job Titles → Most common roles
- Skills in Demand → Required technical skills
- Education vs Salary → Impact of education on salary

Insights

- California and Texas have the highest number of jobs.
- Data Scientist, Analyst, and Software Engineer are top job titles.
- Python, SQL, and Tableau are the most demanded skills.
- Higher education levels lead to higher average salaries.

Recommendations:

- For Job Seekers: Focus on Python, SQL, Tableau; aim for industries like IT, Finance, Biotech.

- For Companies: Recruit in high-demand states and invest in skill development.

Conclusion:

This project demonstrates the complete data analysis lifecycle using Python, Tableau. The findings can help businesses make data-driven decisions in job marketing and useful for job seekers.