Vidyavardhaka College of Engineering

Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi.
Accredited by NBA, New Delhi & NAAC with 'A' Grade
Gokulam 3rd Stage, Mysuru - 570002, Karnataka, India

ADVANCED PYTHON PROGRAMMING LABORATORY
(BISAP317)

Activity Based Assessment Report on

"EXCEL FILE HANDLING "

Submitted by

SINCHANA R NAYAK

4VV22IS101

NISARGA K S

4VV22IS067

Under the guidance of,
Prof.R.Kasturi Rangan
Information science & engineering
Vidyavardhaka college of engineering and technology

Contents

INTRODUCTION:

Handling Excel files in Python is a common task, especially in data analysis and manipulation workflows. Python offers several libraries that facilitate reading, writing, and modifying Excel files. One of the most popular libraries for Excel handling in Python is pandas, which provides high-level data structures and functions designed to make working with structured data fast, easy, and expressive.

Data science involves extracting insights and knowledge from data through various techniques. In this field, roles vary, including data analysts, scientists, and engineers. Salaries depend on factors like experience, location, and industry. Skills in programming, statistical analysis, machine learning, and data visualization are crucial for success in data science careers.

PROBLEM STATEMENT

Explore the job opportunities in data science on Dataset with respect to

i) Representation of job levels

ii) usage of different types of chats to categorize working year of employee, experience level and job count with job categories

iii) List the top 10 paid rolls and average pay list of data science

iv) Develop a visualization using Plotly to illustrate the distribution of employees across different work years

v) Visualize the distribution of employee salaries based on their residence location using a geographical scatter plot.

MOTIVATION

The motivation behind automated Excel handling in Python lies in its ability to streamline data processing, improve accuracy, and support informed decision-making. By leveraging Python's capabilities for data manipulation, analysis, and visualization, organizations can unlock the full potential of their data and gain a competitive edge in today's data-driven world.

DESIGN AND IMPLEMENTATION

DATA SET-1

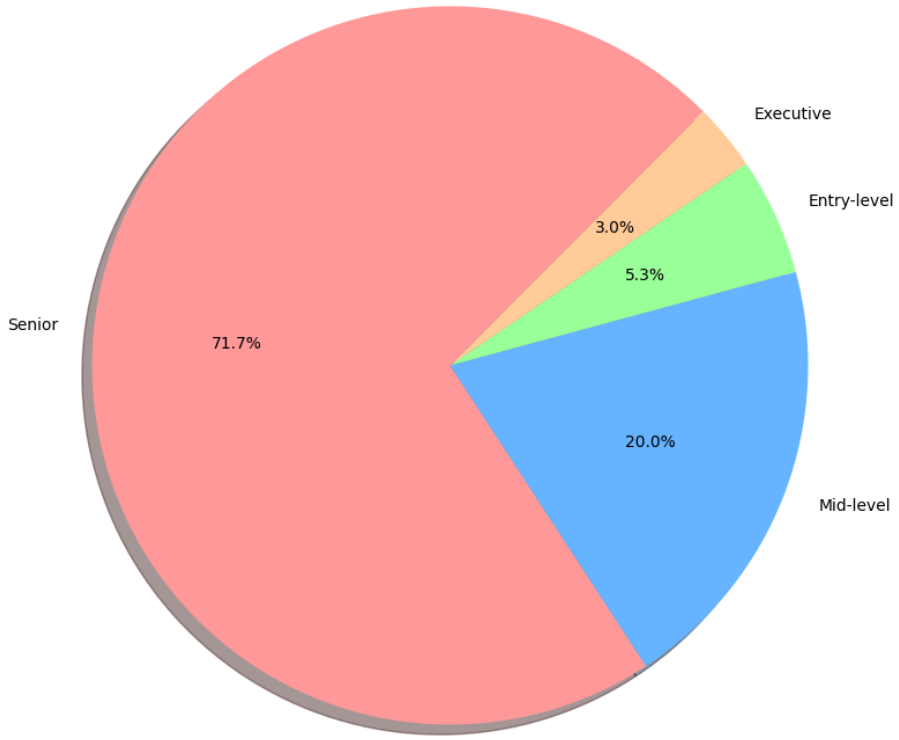
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read
_csv)
import seaborn as sns
import matplotlib.pyplot as plt

import plotly.graph_objs as go
import plotly.express as px
```

```
df=pd.read_csv('python.aba.csv')
df.head()
```

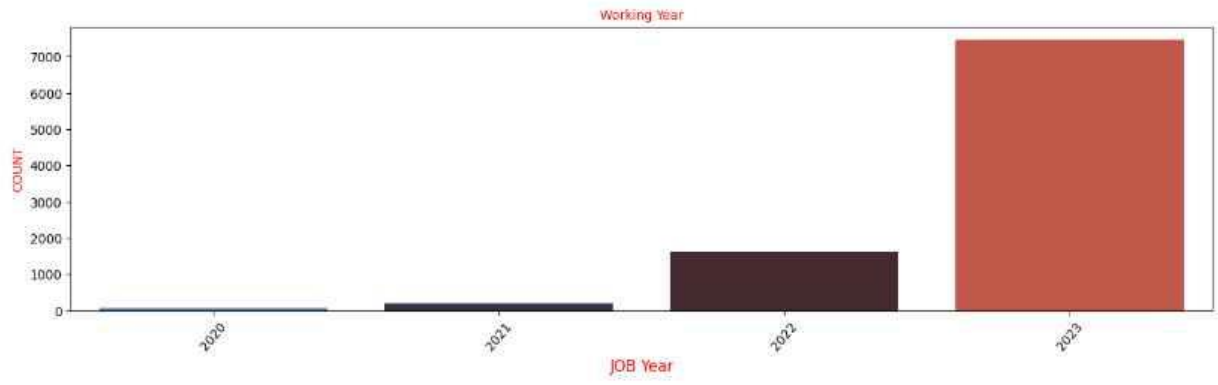
	work_year	job_title	job_category	salary_currency	salary	salary_in_usd	em
0	2023	Data DevOps Engineer	Data Engineering	EUR	88000	95012	Ge
1	2023	Data Architect	Data Architecture and Modeling	USD	186000	186000	Ur
2	2023	Data Architect	Data Architecture and Modeling	USD	81800	81800	Ur
3	2023	Data Scientist	Data Science and Research	USD	212000	212000	Ur
4	2023	Data Scientist	Data Science and Research	USD	93300	93300	Ur

```
labels = df["experience_level"].value_counts().index
sizes = df["experience_level"].value_counts()
colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99',"pink","yello
w"]
plt.figure(figsize = (10,16))
plt.pie(sizes, labels=labels, rotatelabels=False, autopct='%1.1
f%%',colors=colors,shadow=True, startangle=45)
plt.title('States',color = 'red',fontsize = 15)
plt.show()
```

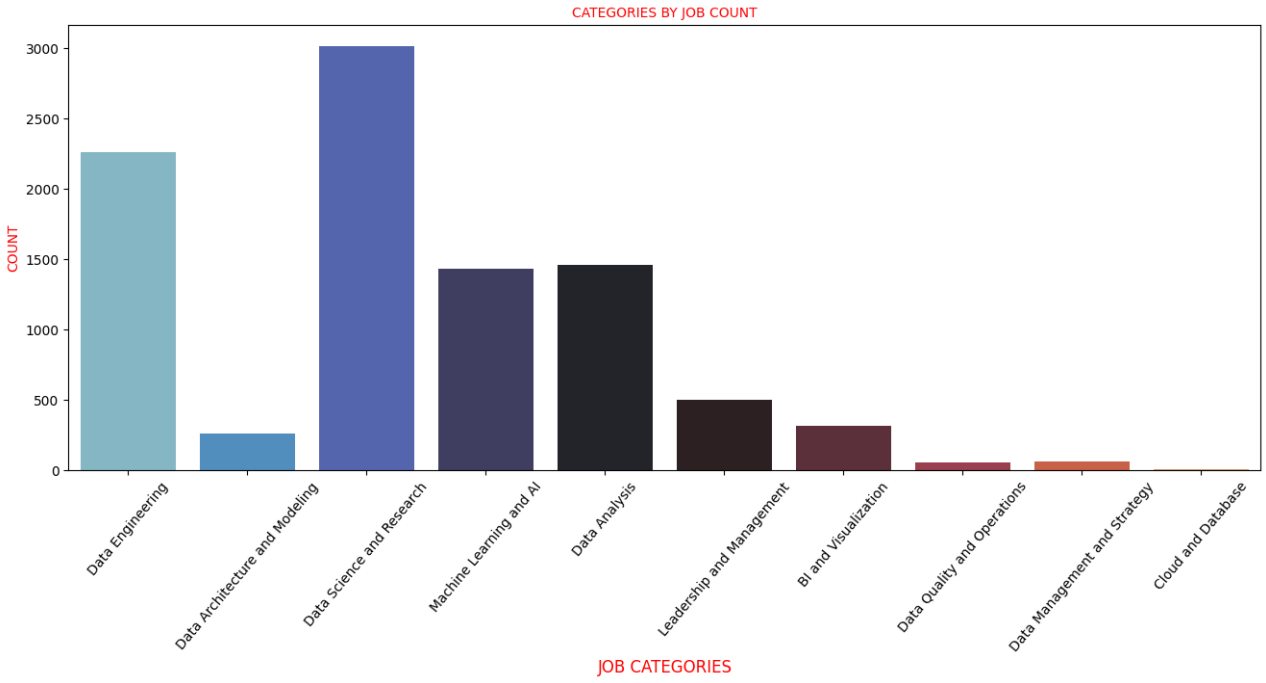
States



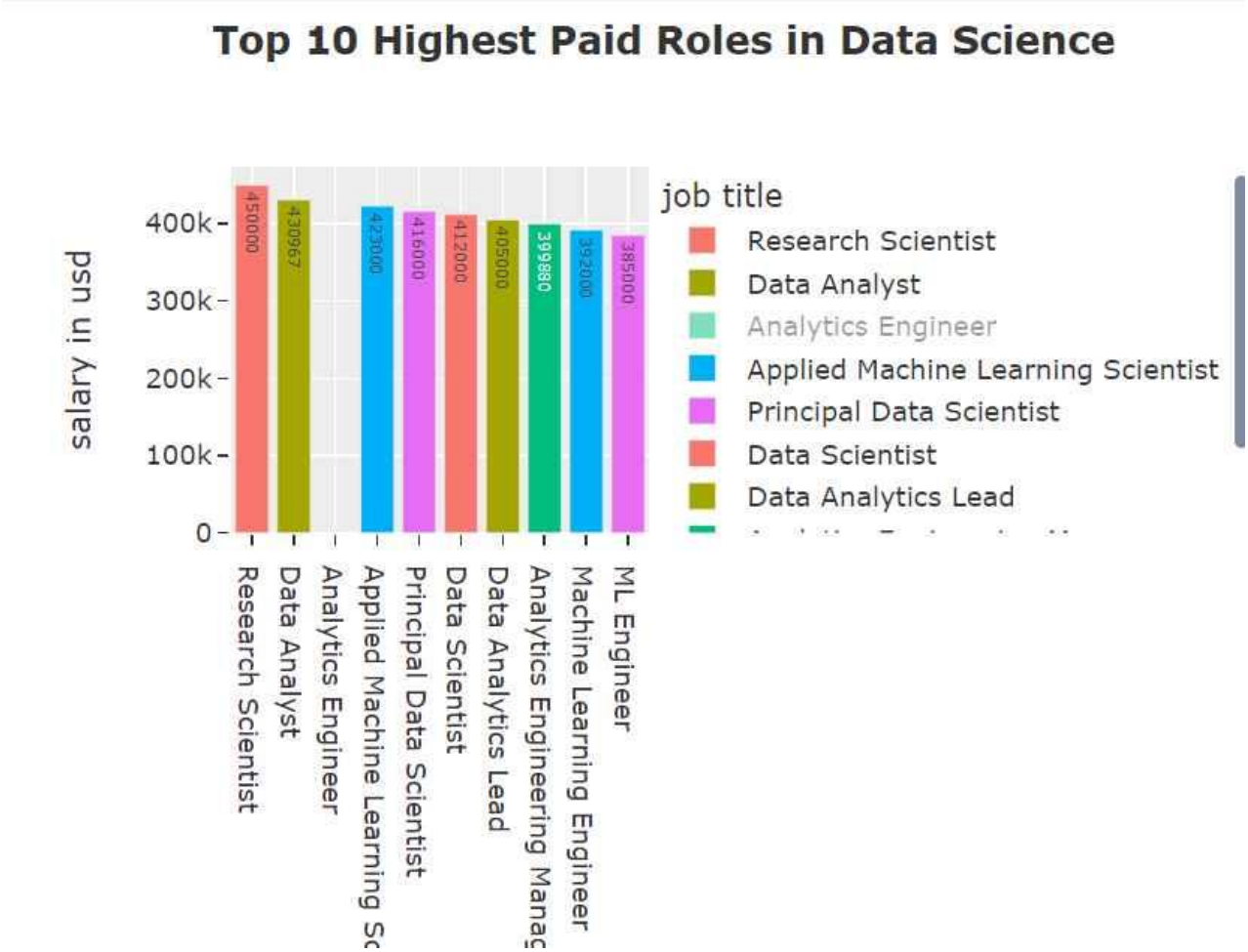

```
plt.figure(figsize=(16,4))
sns.countplot(data=df,x="work_year",palette="icefire")
plt.xticks(fontsize=10,rotation=50)
plt.xlabel("JOB Year",fontsize=12,color="RED")
plt.ylabel("COUNT",fontsize=10,color="RED")
plt.title("Working Year",fontsize=10,color="RED")
plt.show()
```



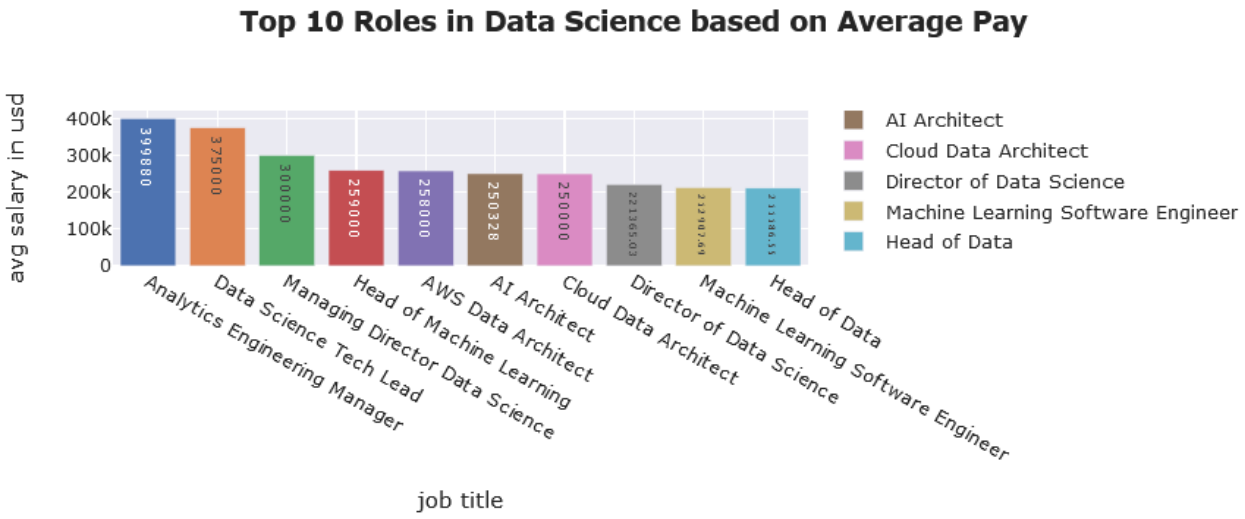
```
plt.figure(figsize=(16,6))
sns.countplot(data=df,x="job_category",palette="icefire")
plt.xticks(fontsize=10,rotation=50)
plt.xlabel("JOB CATEGORIES",fontsize=12,color="RED")
plt.ylabel("COUNT",fontsize=10,color="RED")
plt.title("CATEGORIES BY JOB COUNT",fontsize=10,color="RED")
plt.show()
```



```
fig=px.bar(df.groupby('job_title',as_index=False)['salary_in_usd'].max().sort_values(by='salary_in_usd',ascending=False).head(10),x='job_title',y='salary_in_usd',color='job_title',labels={'job_title':'job title','salary_in_usd':'salary in usd'},template='ggplot2',text='salary_in_usd',title='<b> Top 10 Highest Paid Roles in Data Science')
fig.show()
```

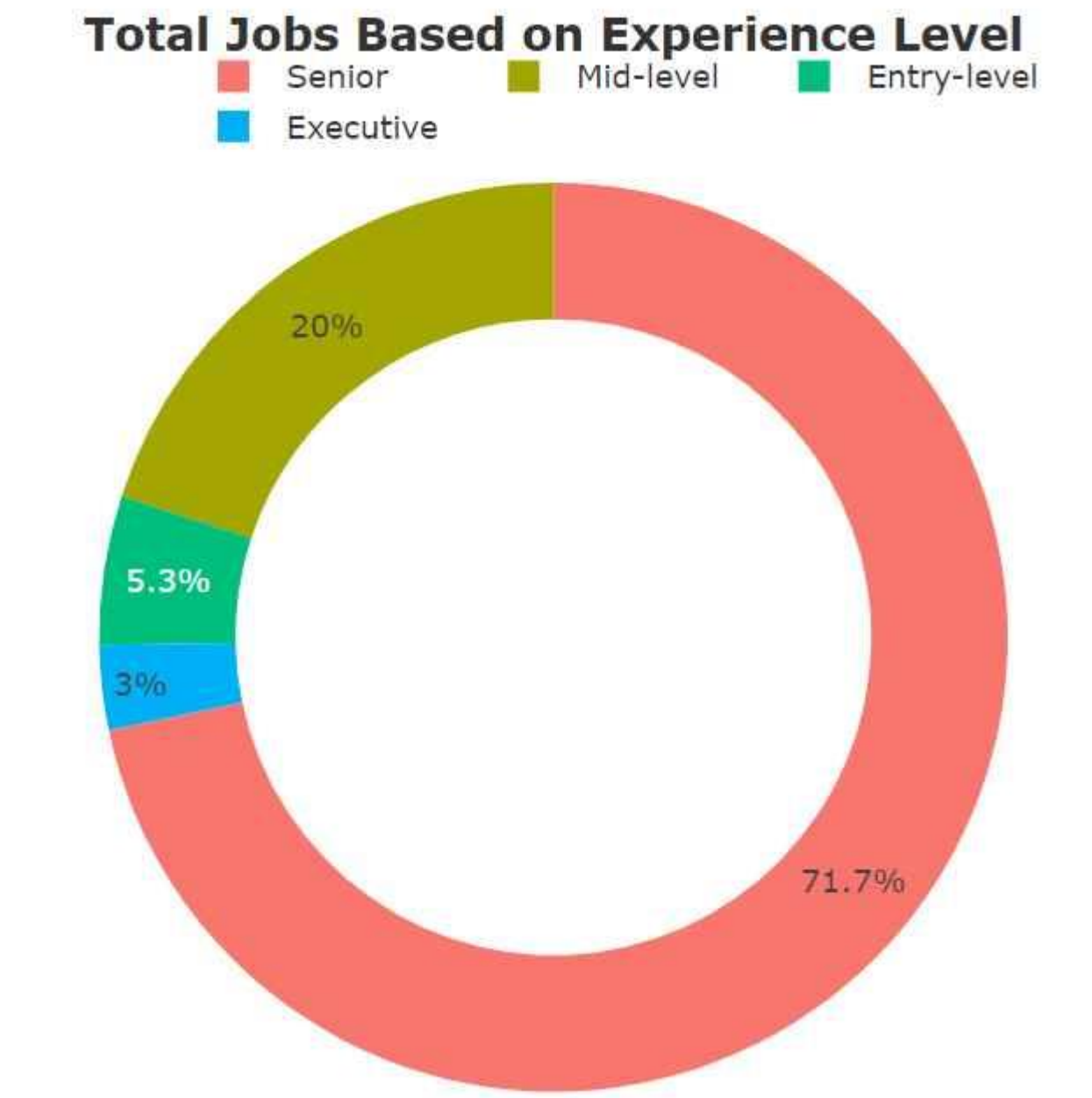



```
z=df.groupby('job_title',as_index=False)['salary_in_usd'].mean().sort_values(by='salary_in_usd',ascending=Fal
z['salary_in_usd']=round(z['salary_in_usd'],2)
fig=px.bar(z.head(10),x='job_title',y='salary_in_usd',color='job_title',labels={'job_title':'job title','sala
fig.update_traces(textfont_size=8)
fig.show()
```

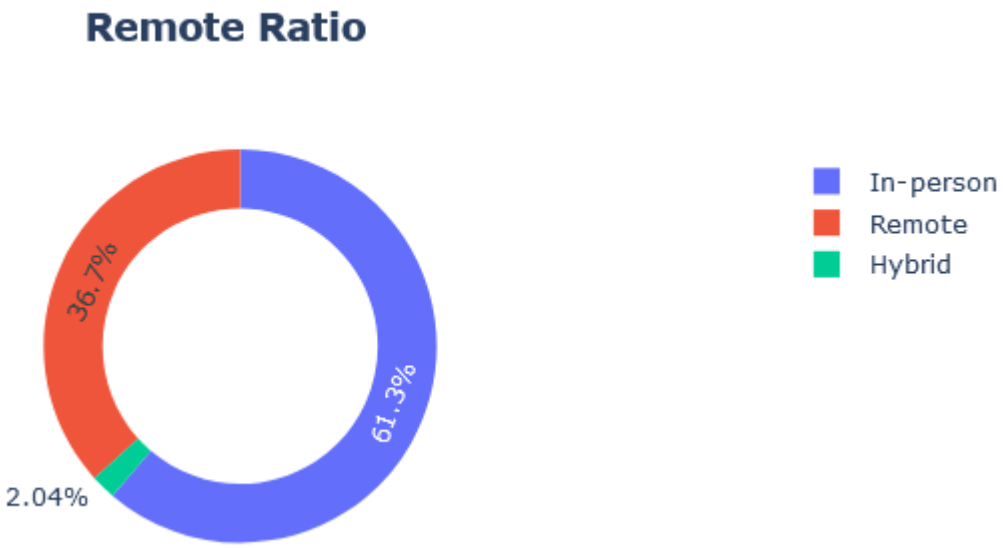



```
px.funnel(df.groupby('company_location',as_index=False)['experie
nce_level'].count().sort_values(by='experience_level',ascending=
False).head(15),y='company_location',x='experience_level',color_
discrete_sequence=['yellow'],labels={'experience_level':'coun
t'},template='seaborn',title='<b> Top 15 Countries having maximu
m Data Science Jobs')
```

```
fig=px.pie(df.groupby('experience_level',as_index=False)['salary
_in_usd'].count().sort_values(by='salary_in_usd',ascending=Fals
e).head(10),names='experience_level',values='salary_in_usd',colo
r='experience_level',hole=0.7,labels={'experience_level':'Experi
ence level ','salary_in_usd':'count'},template='ggplot2',title
='<b>Total Jobs Based on Experience Level')
fig.update_layout(title_x=0.5,legend=dict(orientation='h',yancho
r='bottom',y=1.02,xanchor='right',x=1))
```



```
fig=px.pie(df.groupby('work_setting',as_index=False)['salary_in_usd'].count().sort_values(by='salary_in_usd',
fig.update_layout(title_x=0.5)
```



DATA SET-2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
from plotly.offline import iplot , plot
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings("ignore")
```

```
df=pd.read_csv('python.aba.csv')
df.sample(5)
```

	work_year	job_title	job_category	salary_currency	salary	salary_in_usd
7393	2023	Data Scientist	Data Science and Research	USD	115000	115000
839	2023	Data Scientist	Data Science and Research	USD	127300	127300
4151	2023	Data Engineer	Data Engineering	USD	226600	226600
2825	2023	Data Analyst	Data Analysis	USD	46500	46500
8923	2022	Data Science Manager	Data Science and Research	USD	137141	137141

```
print(f"Number of Row : {df.shape[0]}\nNumber of Columns : {df.shape[1]}")
```

```
Number of Row : 9355
Number of Columns : 12
```

```
# Describe Numiric Data
df.describe().iloc[:,:2]
```

	work_year	salary
count	9355.000000	9355.000000
mean	2022.760449	149927.981293
std	0.519470	63608.835387
min	2020.000000	14000.000000
25%	2023.000000	105200.000000
50%	2023.000000	143860.000000
75%	2023.000000	187000.000000
max	2023.000000	450000.000000

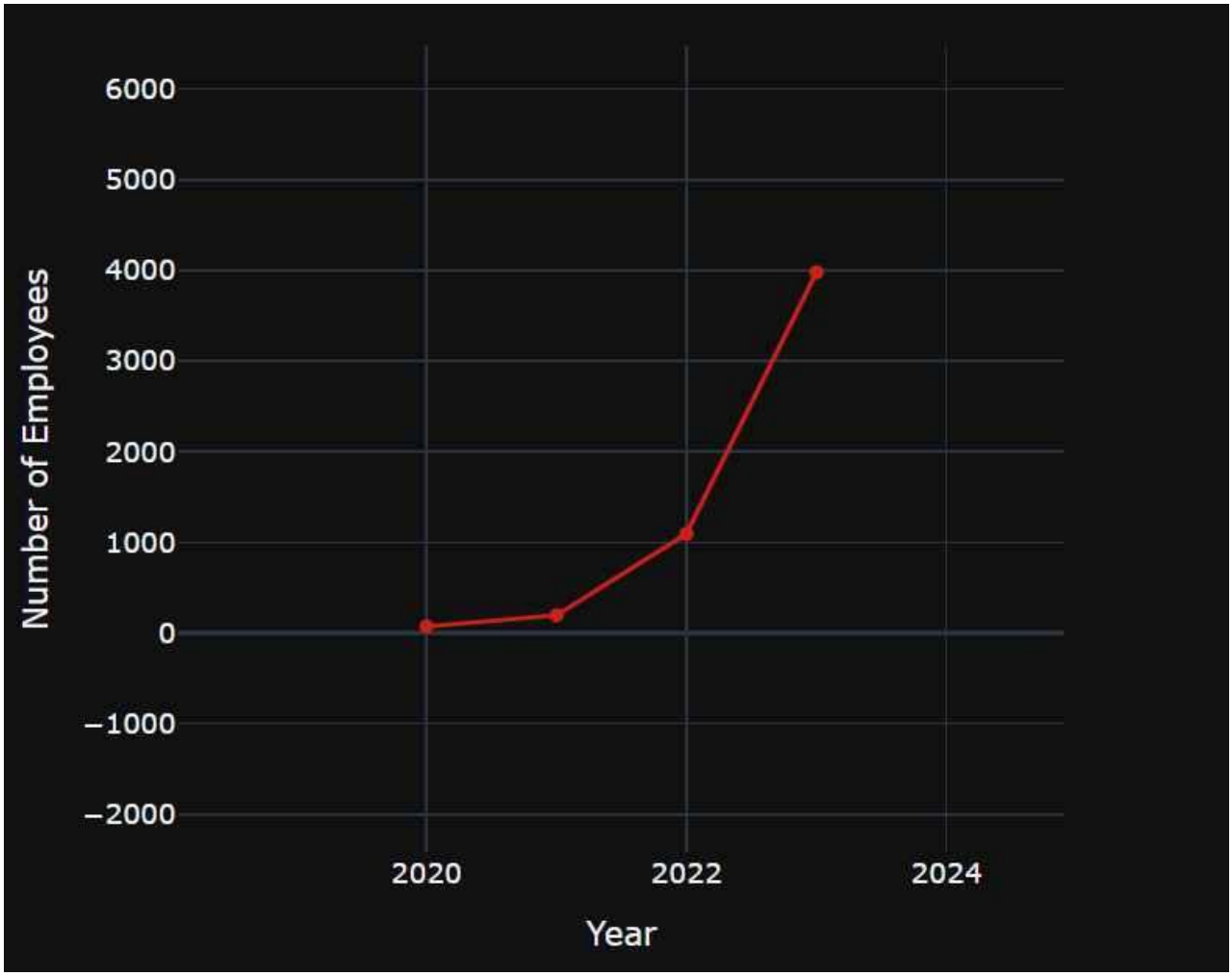
```
# Check Duplicated rows
df.duplicated().any()
```

```
True
```

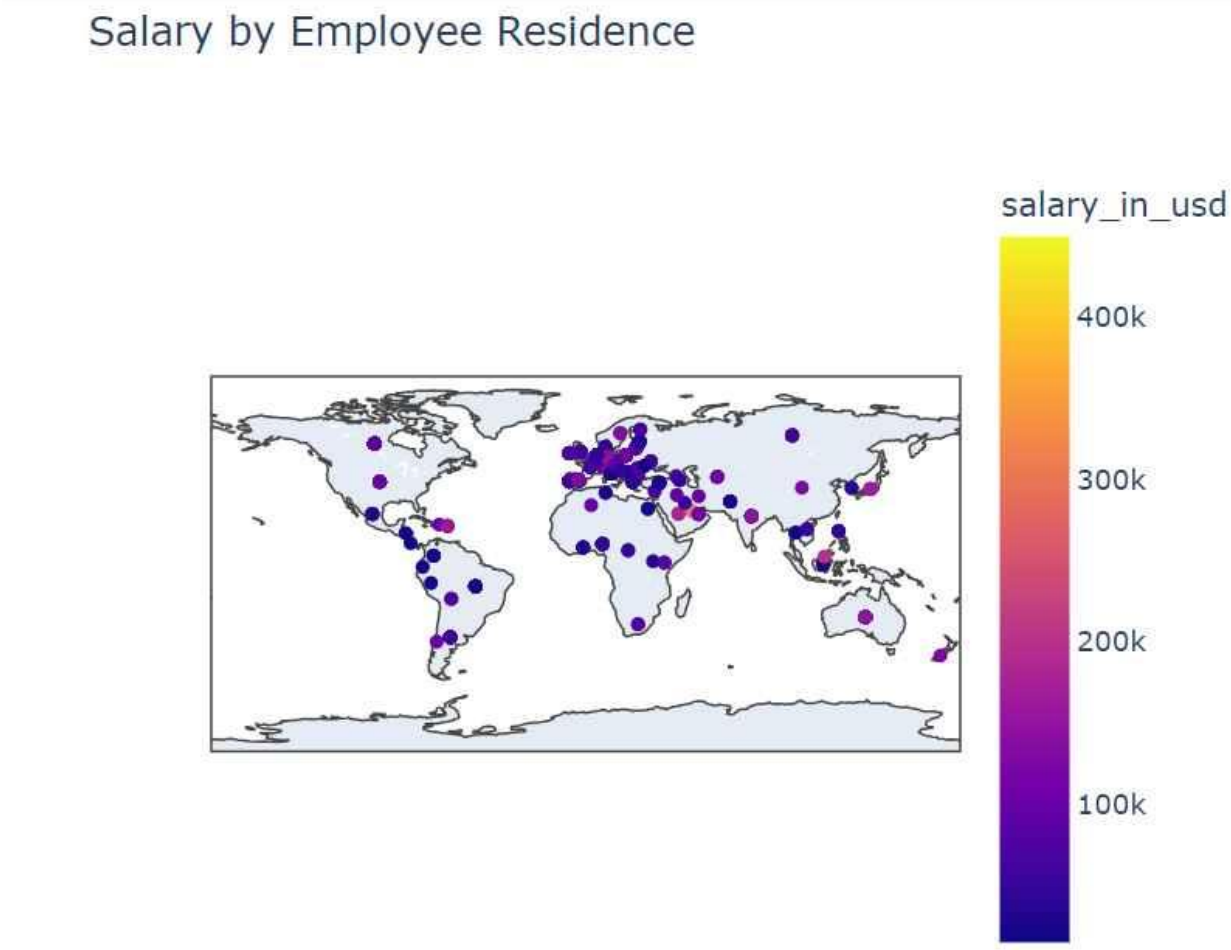
```
print(f"Top Year for Number of Employees '{df['work_year'].value_counts().idxmax()}' with v
alue '{df['work_year'].value_counts().max()}'")
print(f"Least Year for Number of Employees '{df['work_year'].value_counts().idxmin()}' with
value '{df['work_year'].value_counts().min()}'")
```

```
Top Year for Number of Employees '2023' with value '3980'
Least Year for Number of Employees '2020' with value '71'
```

```
iplot(px.line(x = df['work_year'].value_counts().index,
              y = df['work_year'].value_counts().values,
              markers=True,
              labels={'x':'Year','y':'Number of Employees'},
              title='Years of Work',
              line_shape="linear",
              color_discrete_sequence=['#cc2114'],
              template='plotly_dark'
              ))
```

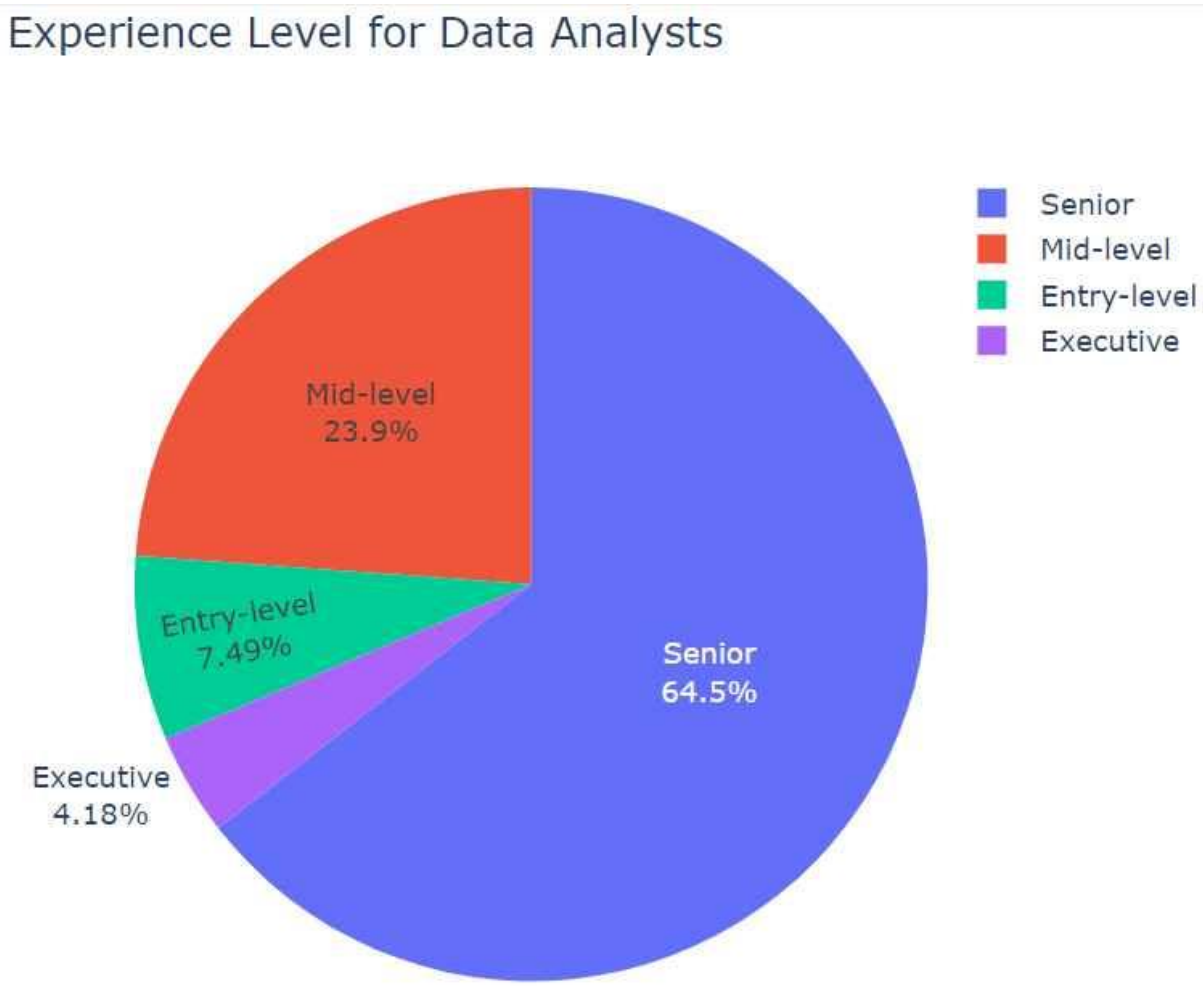


```
iplot(px.scatter_geo(df,
                     locations='employee_residence',
                     locationmode='country names',
                     color='salary_in_usd',
                     hover_name='employee_residence',
                     title='Salary by Employee Residence',
))
```



```
df_experience_level = df['experience_level'].value_counts()
```

```
iplot(px.pie(values=df_experience_level.values,
             names=['Senior','Mid-level','Entry-level','Executiv
e'],
             title='Experience Level for Data Analysts'
).update_traces(textinfo='percent+label'))
```



CONCLUSION

Code demonstrates a comprehensive analysis of job listings in the data field using Python libraries such as pandas, Matplotlib, warnings, plotly, and Seaborn. Here's a summary of key points and findings:

Data Exploration: The initial exploration of the dataset using df.info() and df. describe() provides insights into the structure, data types, and summary statistics of the job listings.

Country-wise Analysis: The analysis identifies the top 10 countries with the highest number of Data Science Jobs, allowing for a focused examination of job opportunities in those regions.

Experience Level Distribution: Understanding the distribution of job listings by experience level sheds light on the skill requirements and expectations of employers in the data field.

Salary Analysis: The analysis of salaries, considering different experience levels and geographical locations, helps in understanding the salary landscape and potential earning opportunities in the data field.

Visualization: The visualizations, including stacked bar plots, line plots, pie chart, and funnel, effectively convey the findings of the analysis and facilitate interpretation by stakeholders.

Overall, the code provides a robust framework for analyzing and visualizing. Salary ranges vary widely depending on various factors including work experience,geographical location,experience level, highest paid jobs in data science,remote ratio of salary.