NAME:- AKSHIT DHAKE

ROLL NO :- 322

PRN:- 202201040177

Topic:-

Read any real-life dataset. Store the data in Data Frames. Identify 10 grains for the given dataset.

	A	B	C	D	E	F	G	H	I
1	work_year	experiNewce_level	employment_type	job_title	salary Rs	salary_in_usd	employee_residNewce	company_location	company_size
2	2023	below10	Full	Principal Data SciNewtist	80000	85847	ES	ES	L
3	2021	above10	Part	ML Newgineer	30000	30000	US	US	S
4	2023	above10	Part	ML Newgineer	25500	25500	US	US	S
5	2023	below10	Full	Data SciNewtist	175000	175000	CA	CA	M
6	2023	below10	Full	Data SciNewtist	120000	120000	CA	CA	M
7	2021	below10	Full	Applied SciNewtist	222200	222200	US	US	L
8	2022	below10	Full	Applied SciNewtist	136000	136000	US	US	L
9	2023	below10	Full	Data SciNewtist	219000	219000	CA	CA	M
10	2023	below10	Full	Data SciNewtist	141000	141000	CA	CA	M
11	2023	below10	Part	Data SciNewtist	147100	147100	US	US	M
12	2021	below10	Full	Data SciNewtist	90700	90700	US	US	M
13	2022	below10	Part	Data Analyst	130000	130000	US	US	M
14	2020	below10	Part	Data Analyst	100000	100000	US	US	M
15	2020	New	Full	Applied SciNewtist	213660	213660	US	US	L
16	2019	New	Part	Applied SciNewtist	130760	130760	US	US	L
17	2019	below10	Full	Data Modeler	147100	147100	US	US	M
18	2020	below10	Full	Data Modeler	90700	90700	US	US	M
19	2023	below10	Part	Data SciNewtist	170000	170000	US	US	M
20	2023	below10	Full	Data SciNewtist	150000	150000	US	US	M
21	2023	above10	Full	Data Analyst	150000	150000	US	US	M
22	2022	above10	Full	Data Analyst	110000	110000	US	US	M
23	2021	below10	Part	Rebelow10arch Newgineer	275000	275000	DE	DE	M
24	2023	below10	Part	Rebelow10arch Newgineer	174000	174000	M	DE	M
25	2022	below10	Part	Analytics Newgineer	230000	230000	GB	GB	

```
import pandas as pd
data = "/content/company.csv"
df = pd.read_csv(data)
print(df)
```

output:-

```
work_year experiNewce_level employment_type                  job_title  \
0        2023           below10             Full  Principal Data SciNewtist
1        2021           above10             Part               ML Newgineer
2        2023           above10             Part               ML Newgineer
3        2023           below10             Full             Data SciNewtist
4        2023           below10             Full             Data SciNewtist
5        2021           below10             Full          Applied SciNewtist
6        2022           below10             Full          Applied SciNewtist
7        2023           below10             Full             Data SciNewtist
8        2023           below10             Full             Data SciNewtist
9        2023           below10             Part             Data SciNewtist
10       2021           below10             Full             Data SciNewtist
11       2022           below10             Part                Data Analyst
12       2020           below10             Part                Data Analyst
13       2020               New             Full          Applied SciNewtist
14       2019               New             Part          Applied SciNewtist
15       2019           below10             Full                Data Modeler
16       2020           below10             Full                Data Modeler
17       2023           below10             Part             Data SciNewtist
18       2023           below10             Full             Data SciNewtist
19       2023           above10             Full                Data Analyst
20       2022           above10             Full                Data Analyst
21       2021           below10             Part    Rebelow10arch Newgineer
22       2023           below10             Part    Rebelow10arch Newgineer
23       2022           below10             Part        Analytics Newgineer

    salary Rs  salary_in_usd employee_residNewce company_location
company_size
0       80000          85847                  ES                ES
L
1       30000          30000                  US                US
S
2       25500          25500                  US                US
S
3      175000         175000                  CA                CA
M
4      120000         120000                  CA                CA
M
5      222200         222200                  US                US
L
6      136000         136000                  US                US
L
7      219000         219000                  CA                CA
M
```

```
8      141000         141000                   CA               CA
M
9      147100         147100                   US               US
M
10      90700          90700                   US               US
M
11     130000         130000                   US               US
M
12     100000         100000                   US               US
M
13     213660         213660                   US               US
L
14     130760         130760                   US               US
L
15     147100         147100                   US               US
M
16      90700          90700                   US               US
M
17     170000         170000                   US               US
M
18     150000         150000                   US               US
M
19     150000         150000                   US               US
M
20     110000         110000                   US               US
M
21     275000         275000                   DE               DE
M
22     174000         174000                    M               DE
M
23     230000         230000                   GB               GB
NaN
```

qurries:-

```
# 1) ifnd which country has maximun number of companies
# Count the occurrences of each country in the 'country' column
country_counts = df['company_location'].value_counts()

# Get the country with the maximum occurrences
max_country = country_counts.idxmax()

# Print the country with the maximum occurrences
print("Country with the maximum occurrences:", max_country)
```

output:- Country with the maximum occurrences: US

```
# 2) Find the maximum occurring job title
most_common_job_title = df['job_title'].value_counts().idxmax()

# Print the most common job title
print("Most common job title:", most_common_job_title)
```

Output:- Most common job title: Data SciNewtist

```
# 3 TO find which size of company has most full job
full_employment_df = df[df['employment_type'] == 'Full']

# Count the occurrences of each company size in the filtered DataFrame
company_size_counts = full_employment_df['company_size'].value_counts()

# Find the company size with the most "full" employment type
most_common_company_size = company_size_counts.idxmax()

# Print the company size with the most "full" employment type
print("Company size with the most 'full  time ' employment type:",
most_common_company_size)
```

Output:- ompany size with the most 'full time ' employment type: M

```
# 4) which job title has maximum salary

# Convert the salary column to numeric values
df['salary Rs'] = df['salary Rs'].replace('Rs', '',
regex=True).astype(float)

# Find the job title with the highest salary
highest_salary_job_title = df.loc[df['salary Rs'].idxmax(), 'job_title']

# Print the job title with the highest salary
print("Job title with the highest salary:", highest_salary_job_title
```

Output:- Job title with the highest salary: Rebelow10arch Newgineer

```
# 5) job title with least salary
# Sort the DataFrame by salary in ascending order
sorted_df = df.sort_values(by='salary Rs')

# Get the job title with the least salary
least_salary_job_title = sorted_df['job_title'].iloc[0]
```

```
# Get the value of the least salary
least_salary_value = sorted_df['salary Rs'].iloc[0]

# Print the job title with the least salary and its value
print("Job title with the least salary:", least_salary_job_title)
print("Value of the least salary:", least_salary_value)
```

Output:- Job title with the least salary: ML Newgineer

Value of the least salary: 25500.0

```
# 6) To find which employee does not reside in same country of there
company and this details

# Filter the DataFrame for employees who work in a different country
different_country_employees = df[df['company_location'] !=
df['employee_residNewce']]

# Print the employees who work in a different country
print("Employees who work in a different country:")
print(different_country_employees)
```

Output:-

```
Employees who work in a different country:

    work_year experiNewce_level employment_type                job_title
\
22       2023           below10            Part  Rebelow10arch Newgineer

    salary Rs  salary_in_usd employee_residNewce company_location
company_size
22   174000.0         174000                   M               DE
M
```

```
# 7) To find which country has maximum number of Applied SciNewtist
employees

# Filter the DataFrame for employees with the job title "Applied
Scientist"
Applied_SciNewtist_employees = df[df['job_title'] == 'Applied SciNewtist']

# Count the occurrences of each country in the filtered DataFrame
```

```
country_counts =
Applied_SciNewtist_employees['company_location'].value_counts()

# Find the country with the maximum number of "Applied Scientist"
employees
country_with_max_employees = country_counts.idxmax()

# Print the country with the maximum number of "Applied Scientist"
employees
print("Country with the maximum number of 'Applied SciNewtist employees:",
country_with_max_employees)
```

Output:- Country with the maximum number of 'Applied SciNewtist employees: US

```
 # 8) to find which job title occur least time

# Find the minimum occurrence of job titles
min_occurrence = df['job_title'].value_counts().min()

# Find the job title(s) with the minimum occurrence
min_occurrence_job_titles =
df['job_title'].value_counts()[df['job_title'].value_counts() ==
min_occurrence]

# Print the minimum occurrence and the corresponding job title(s)
print("Minimum Occurrence:", min_occurrence)
print("Job Title(s) with Minimum Occurrence:")
print(min_occurrence_job_titles)
```

Ouput:-

```
Minimum Occurrence: 1

Job Title(s) with Minimum Occurrence:
Principal Data SciNewtist    1
Analytics Newgineer          1
Name: job_title, dtype: int64
```

```
# 9) to find which job tile has full time job and how many of them have
# Filter the DataFrame for rows where employment_type is "Full"
full_time_jobs = df[df['employment_type'] == 'Full']

# Count the occurrences of job titles for full-time jobs
```

```
job_title_counts = full_time_jobs['job_title'].value_counts()

# Get the job title with the most full-time occurrences and its count
most_common_job_title = job_title_counts.idxmax()
most_common_job_title_count = job_title_counts.max()

# Print the job title with the most full-time occurrences and its count
print("Job Title with the Most Full-Time job:", most_common_job_title)
print("Count:", most_common_job_title_count)
```

Ouput:-

```
Job Title with the Most Full-Time job: Data SciNewtist

Count: 6
```

```
#  10 )To find details of the part time employee who has maximum salary

# Filter the DataFrame for rows where employment_type is "Part"
part_time_jobs = df[df['employment_type'] == 'Part']

# Find the maximum salary among part-time jobs
max_salary = part_time_jobs['salary Rs'].max()

# Retrieve the details of the job with the maximum salary
max_salary_job = part_time_jobs[part_time_jobs['salary Rs'] == max_salary]

# Print the maximum salary and its details
print("Maximum Salary (Part-Time):", max_salary)
print("Job Details:")
print(max_salary_job)
```

Ouput:-

Maximum Salary (Part-Time): 275000.0
Job Details:

	work_year	experiNewce_level	employment_type	job_title	\
21	2021	below10	Part	Rebelow10arch Newgineer	

	salary Rs	salary_in_usd	employee_residNewce	company_location	company_size
21	275000.0	275000	DE	DE	M
