```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 %matplotlib inline
```

```
1 df = pd.read_csv('/content/501+Case1+Dataset (4).csv')
2
```

```
1 df
```

	Employee_Name	EmpID	Sex	GenderID	MaritalDesc	MarriedID	MaritalStatusID	DOB	State	Zip	...	ManagerID	RecruitmentSour
0	Le, Binh	10232	F	0	Single	0	0	06/14/87	MA	1886	...	13.0	Inde
1	Martin, Sandra	10110	F	0	Single	0	0	11/07/1987	MA	2135	...	10.0	Google Sea
2	Myers, Michael	10216	M	1	Single	0	0	04/18/80	MA	1550	...	20.0	Linke
3	Navathe, Kurt	10079	M	1	Single	0	0	04/25/70	MA	2056	...	13.0	Inde
4	Sutwell, Barbara	10209	F	0	Single	0	0	08/15/68	MA	2718	...	16.0	Inde
...	...	...	...	...	...	...	...	...	...	...	...	...	
306	Wilkes, Annie	10204	F	0	Divorced	0	2	07/30/83	MA	1876	...	19.0	Google Sea
307	Demita, Carla	10100	F	0	Separated	0	3	02/25/51	MA	2343	...	18.0	Google Sea
308	Lundy, Susan	10096	F	0	Widowed	0	4	12/26/76	MA	2122	...	22.0	Linke
309	MacLennan, Samuel	10191	M	1	Widowed	0	4	11/09/1972	MA	1938	...	11.0	Inde
310	Thibaud, Kenneth	10268	M	1	Widowed	0	4	09/16/75	MA	2472	...	39.0	Ot

311 rows × 33 columns

```
1 #1 Import the data & check the head, tail for it ?
2 df. head()
```

	Employee_Name	EmpID	Sex	GenderID	MaritalDesc	MarriedID	MaritalStatusID	DOB	State	Zip	...	ManagerID	RecruitmentSource
0	Le, Binh	10232	F	0	Single	0	0	06/14/87	MA	1886	...	13.0	Indeec
1	Martin, Sandra	10110	F	0	Single	0	0	11/07/1987	MA	2135	...	10.0	Google Search
2	Myers, Michael	10216	M	1	Single	0	0	04/18/80	MA	1550	...	20.0	LinkedIr
3	Navathe, Kurt	10079	M	1	Single	0	0	04/25/70	MA	2056	...	13.0	Indeec
4	Sutwell, Barbara	10209	F	0	Single	0	0	08/15/68	MA	2718	...	16.0	Indeec

5 rows × 33 columns

```
1 df.tail
```

```
#2 Check the shape, size of the data ?
df.shape
```

```
(311, 33)
```

```
df .size
```

```
10263
```

Kenneth

```
df .info
```

```
<bound method DataFrame.info of          Employee_Name  EmpID Sex  GenderID MaritalDesc  MarriedID  \
0             Le, Binh  10232   F         0      Single          0
1       Martin, Sandra  10110   F         0      Single          0
2       Myers, Michael  10216  M          1      Single          0
3        Navathe, Kurt  10079  M          1      Single          0
4     Sutwell, Barbara  10209   F         0      Single          0
..                 ...    ...  ..       ...         ...        ...
306      Wilkes, Annie  10204   F         0    Divorced          0
307      Demita, Carla  10100   F         0   Separated          0
308       Lundy, Susan  10096   F         0     Widowed          0
309  MacLennan, Samuel  10191  M          1     Widowed          0
310   Thibaud, Kenneth  10268  M          1     Widowed          0

     MaritalStatusID         DOB State   Zip  ... ManagerID RecruitmentSource  \
0                  0    06/14/87    MA  1886  ...      13.0            Indeed
1                  0  11/07/1987    MA  2135  ...      10.0     Google Search
2                  0    04/18/80    MA  1550  ...      20.0          LinkedIn
3                  0    04/25/70    MA  2056  ...      13.0            Indeed
4                  0    08/15/68    MA  2718  ...      16.0            Indeed
..               ...         ...   ...   ...  ...       ...               ...
306                2    07/30/83    MA  1876  ...      19.0     Google Search
307                3    02/25/51    MA  2343  ...      18.0     Google Search
308                4    12/26/76    MA  2122  ...      22.0          LinkedIn
309                4  11/09/1972    MA  1938  ...      11.0            Indeed
310                4    09/16/75    MA  2472  ...      39.0             Other

    LastPerformanceReview_Date PerformanceScore PerfScoreID EngagementSurvey  \
0                   01/08/2019      Fully Meets           3             4.10
1                    1/14/2019      Fully Meets           3             4.50
2                    1/22/2019      Fully Meets           3             4.10
3                    2/25/2019      Fully Meets           3             5.00
4                    1/31/2019      Fully Meets           3             3.40
..                         ...              ...         ...              ...
306                 02/06/2011      Fully Meets           3             3.60
307                 05/06/2015      Fully Meets           3             4.62
308                 06/10/2016      Fully Meets           3             4.65
309                 04/01/2017      Fully Meets           3             3.08
310                  7/14/2010      Fully Meets           3             4.10

     EmpSatisfaction SpecialProjectsCount  DaysLateLast30  Absences
0                  5                    7               0         2
1                  5                    4               0        14
2                  4                    0               0        13
3                  3                    6               0        17
4                  5                    0               0        13
..               ...                  ...             ...       ...
306                5                    0               0         9
307                5                    0               0         1
308                4                    0               0        15
309                4                    0               0        18
310                4                    0               0        15

[311 rows x 33 columns]>
```

```
#3.How many columns have categorical features ?
len(df.select_dtypes(include="object").columns)
```

```
17
```

```
4 How many unique values are present in RaceDesc column ?
df['Racedesc'].unique()
```

```
  File "<ipython-input-26-4a7ba73c9cdf>", line 1
    4 How many unique values are present in RaceDesc column?
      ^
SyntaxError: invalid syntax
```

```
#5 Check for mean, max .value, min .value, count, standard deviation of ManagerID
column ?
df['ManagerID'].describe()
```

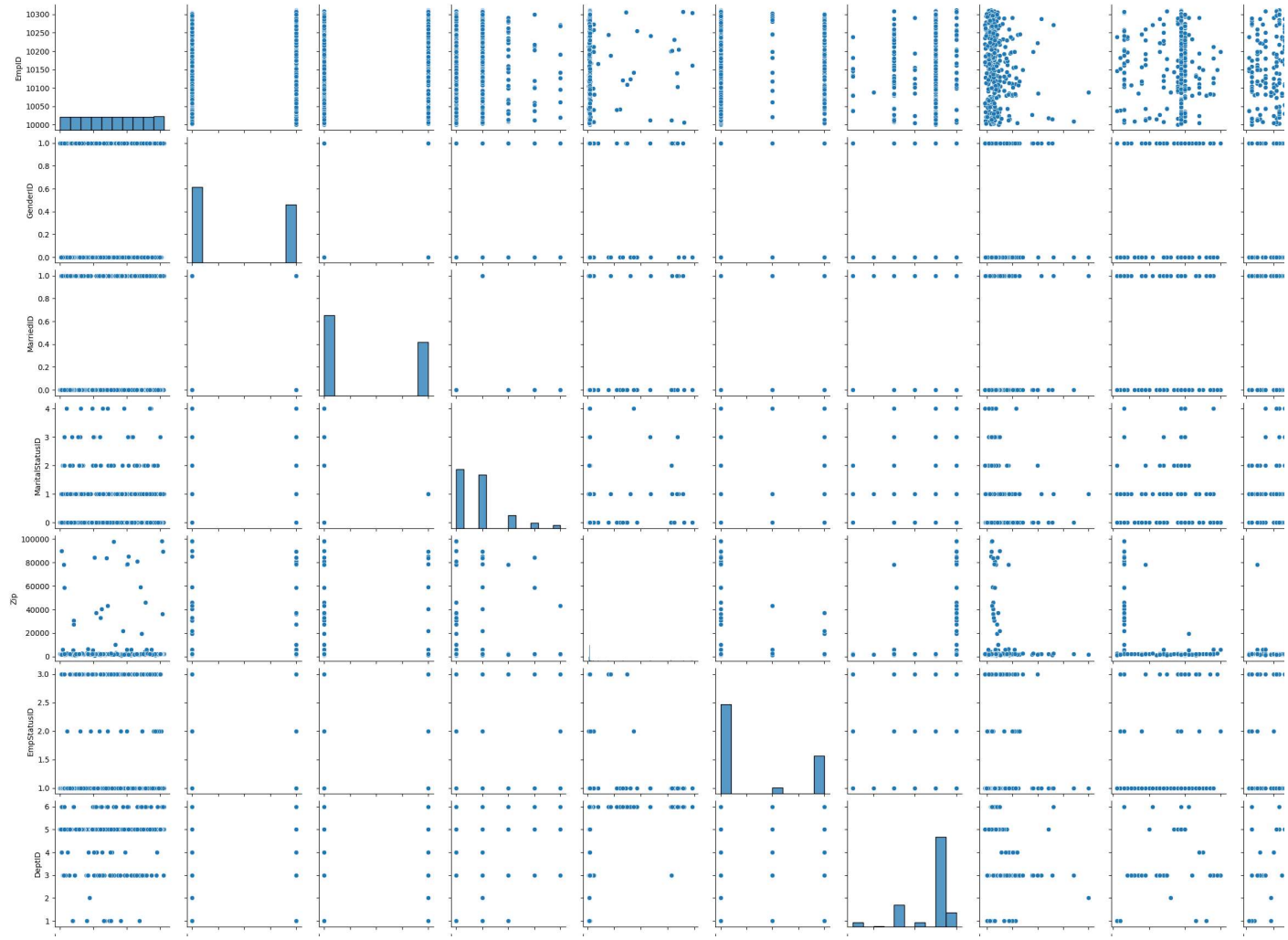
```
count    303.000000
mean      14.570957
std        8.078306
min        1.000000
25%       10.000000
50%       15.000000
75%       19.000000
max       39.000000
Name: ManagerID, dtype: float64
```

```
#6 Count the no of categorical , numerical columns ?
len(df.select_dtypes(exclude="object").columns)
```

```
16
```

```
# 7 Make a diversity report about the dataset?
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(13,7))
sns.pairplot(data=df)

```

```
<seaborn.axisgrid.PairGrid at 0x7fe6339fc760>
<Figure size 1300x700 with 0 Axes>
```




```
#8 Which columns have correlation with each other, what will you interpret from it?
df.corr()

```

```
<ipython-input-5-2db1fe21e460>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version,
  df.corr()
```

```
#9 Add a new column which specifies the number of characters in Employee_Name
column?
df['employee_char'] = df['Employee_Name'].str.len()
df.head(2)
```

	Employee_Name	EmpID	Sex	GenderID	MaritalDesc	MarriedID	MaritalStatusID	DOB	State	Zip	...	RecruitmentSource	LastPerfc
0	Le, Binh	10232	F	0	Single	0	0	06/14/87	MA	1886	...	Indeed	
1	Martin, Sandra	10110	F	0	Single	0	0	11/07/1987	MA	2135	...	Google Search	

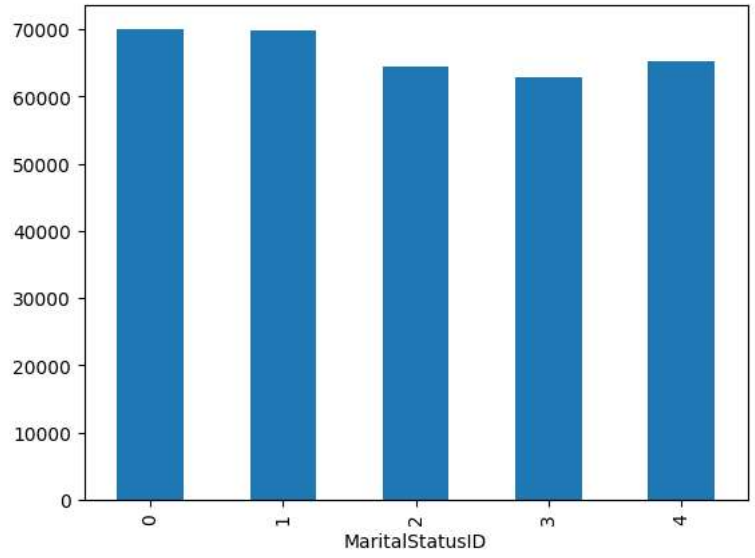
2 rows × 34 columns

```
#10 Round up the values in EngagementSurvey column ?
df['EngagementSurvey'].round()


```

```
0      4.0
1      4.0
2      4.0
3      5.0
4      3.0
      ...
306    4.0
307    5.0
308    5.0
309    3.0
310    4.0
Name: EngagementSurvey, Length: 311, dtype: float64
```

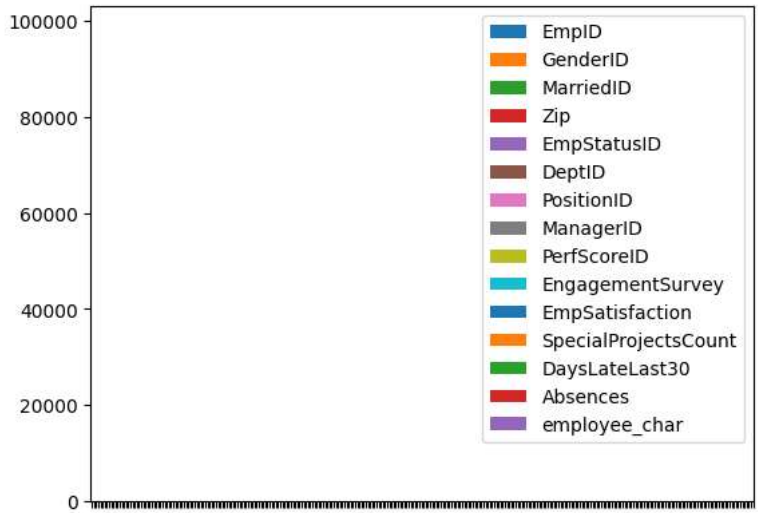
```
#11 a. Does marital status have any impact on salary?
df.groupby(['MaritalStatusID'])['Salary'].mean().plot(kind='bar')
```

```
<Axes: xlabel='MaritalStatusID'>
```



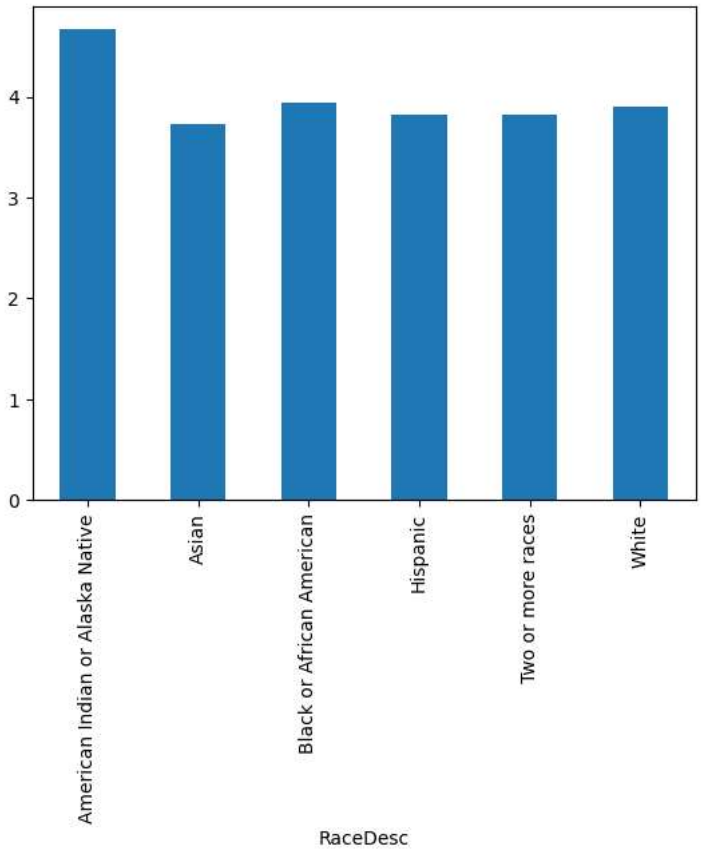

```
#11

df.groupby(['MaritalStatusID','Salary']).mean().plot(kind='bar')
```

```
<ipython-input-48-d32890a7ede7>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future v
  df.groupby(['MaritalStatusID','Salary']).mean().plot(kind='bar')
<Axes: xlabel='MaritalStatusID,Salary'>
```




```
##11 (b)Does RaceDesc have any impact on EmpSatisfaction?
df.groupby(['RaceDesc'])['EmpSatisfaction'].mean().plot(kind='bar')
```

```
<Axes: xlabel='RaceDesc'>
```




```
##11 (3)
```

```
## CASE 2
```

```
db = pd.read_csv('//content/502+Case2+Dataset.csv')
```

```
db.columns
```

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
```

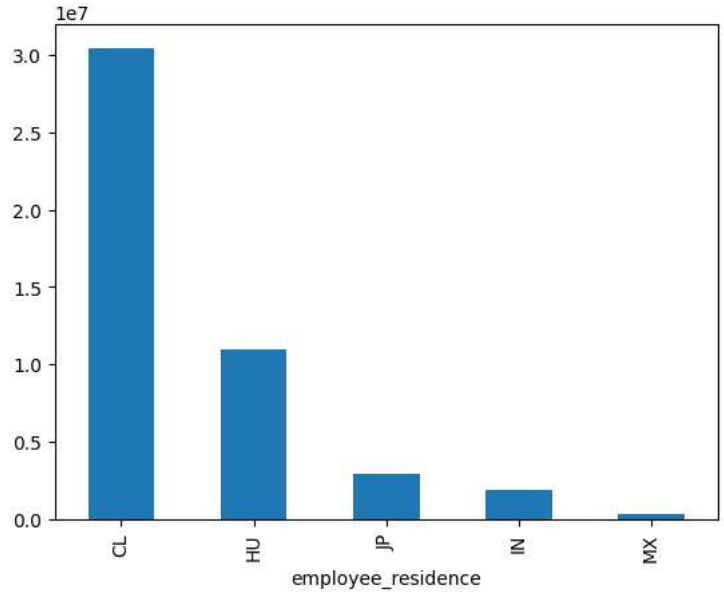
```
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size'],
      dtype='object')
```

```
db.head(5)
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_res
0	0	2020	MI	FT	Data Scientist	70000	EUR	79833	
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	109024	
3	3	2020	MI	FT	Product Data Analyst	20000	USD	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	150000	

```
##1 How does where you live affect your salary?

db.groupby(['employee_residence'])['salary'].mean().sort_values(ascending=False).head().plot(kind='bar')
```

```
<Axes: xlabel='employee_residence'>
```




```
#2 How has the demand of the jobs been throughout the years?
db['work_year'].value_counts()

```

```
2022    318
2021    217
2020     72
Name: work_year, dtype: int64
```

```
##3 How common is to work remote?
db['remote_ratio'].value_counts()

```

```
100    381
0      127
50      99
Name: remote_ratio, dtype: int64
```

```
##4 What the highest paying jobs with entry level as well as for senior level experienced?

db[(db['experience_level'] == "EN") | (db['experience_level'] == 'SE')]

db.groupby(['experience_level'])['salary'].max()

```

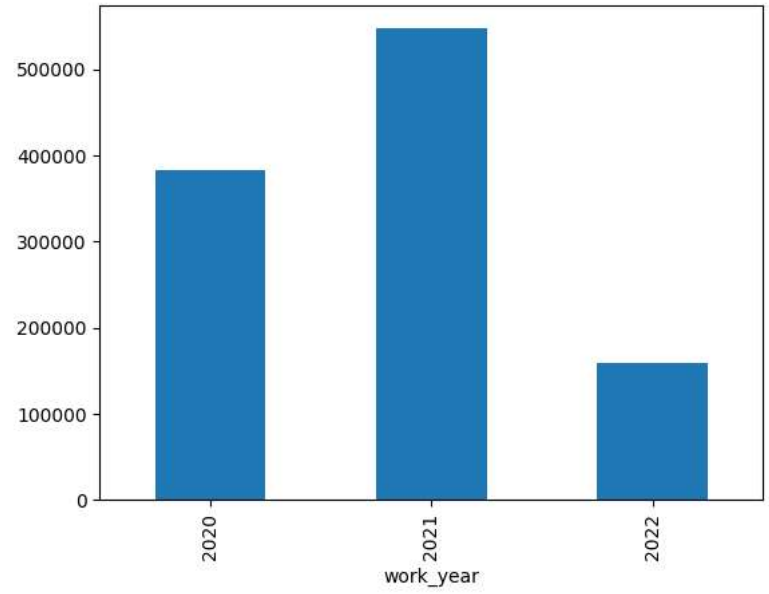
```
experience_level
EN     4450000
EX     6000000
MI    30400000
SE     7000000
Name: salary, dtype: int64
```

```
##5 What company size hire the most?
db['company_size'].value_counts().head()

```

```
M    326
L    198
S     83
Name: company_size, dtype: int64
```

```
##6 How has average salary changed throughout the years?
db.groupby(['work_year'])['salary'].mean().plot(kind='bar')

```

```
<Axes: xlabel='work_year'>
```



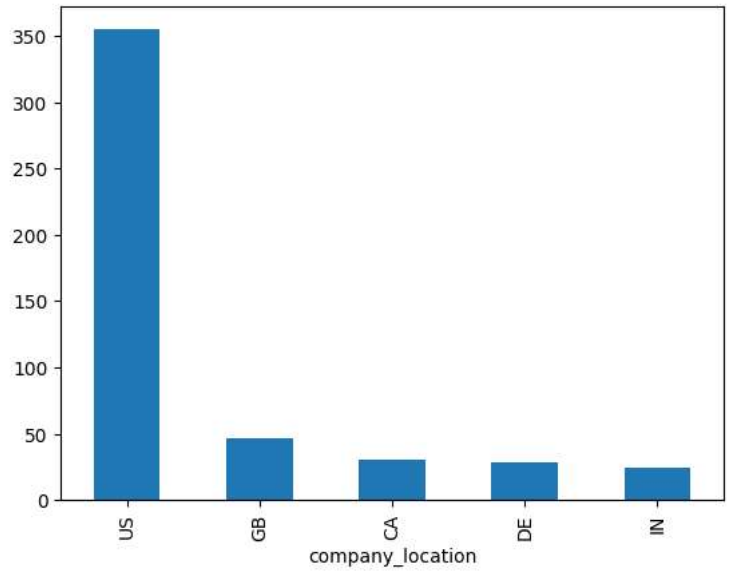

```
##7 What are most popular roles in Data Science ?

db['job_title'].value_counts()
```

```
Data Scientist                            143
Data Engineer                             132
Data Analyst                               97
Machine Learning Engineer                  41
Research Scientist                         16
Data Science Manager                       12
Data Architect                             11
Big Data Engineer                           8
Machine Learning Scientist                  8
Principal Data Scientist                    7
AI Scientist                                7
Data Science Consultant                     7
Director of Data Science                    7
Data Analytics Manager                      7
ML Engineer                                 6
Computer Vision Engineer                    6
BI Data Analyst                             6
Lead Data Engineer                          6
Data Engineering Manager                    5
Business Data Analyst                       5
Head of Data                                5
Applied Data Scientist                      5
Applied Machine Learning Scientist          4
Head of Data Science                        4
Analytics Engineer                          4
Data Analytics Engineer                     4
```

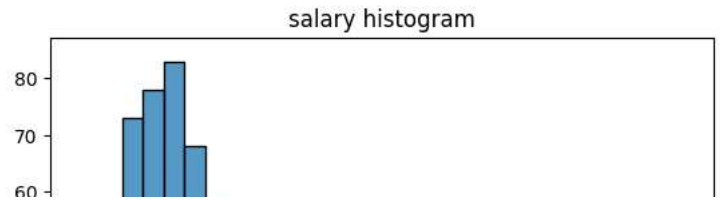
```
Machine Learning Developer                  3
Machine Learning Infrastructure Engineer    3
Lead Data Scientist                         3
Computer Vision Software Engineer           3
Lead Data Analyst                           3
Data Science Engineer                       3
Principal Data Engineer                     3
Principal Data Analyst                      2
ETL Developer                               2
Product Data Analyst                        2
Director of Data Engineering                2
Financial Data Analyst                      2
Cloud Data Engineer                         2
Lead Machine Learning Engineer              1
NLP Engineer                                1
Head of Machine Learning                    1
3D Computer Vision Researcher               1
Data Specialist                             1
Staff Data Scientist                        1
Big Data Architect                          1
Finance Data Analyst                        1
Marketing Data Analyst                      1
Machine Learning Manager                    1
Data Analytics Lead                         1
Name: job_title, dtype: int64
```

```
##8 Which country hire the most people in Data Science?
db.groupby(['company_location'])['job_title'].count().sort_values(ascending=False).head().plot(kind='bar')
```

```
<Axes: xlabel='company_location'>
```




```
##9 What is the distribution of Salaries?

sns.histplot(db['salary_in_usd'])
plt.title('salary histogram')
plt.show()
```




```
##10. How much can you expect depending on your years of experience?
db.groupby(['experience_level'])['salary'].mean()

```

```
experience_level
EN    264622.454545
EX    427072.115385
MI    480617.690141
SE    213949.353571
Name: salary, dtype: float64
```

```
##11.Which year do people prefer to stay at home the most?
db.groupby(['remote_ratio'])['work_year'].value_counts()
```

```
remote_ratio  work_year
0             2022          78
              2021          34
              2020          15
50            2021          66
              2020          21
              2022          12
100           2022         228
              2021         117
              2020          36
Name: work_year, dtype: int64
```

```
##12. Which country has the highest pay?
db.groupby(['company_location'])['salary'].max().head(5)

```

```
company_location
AE     120000
AS    1335000
AT      80000
AU     150000
BE      75000
Name: salary, dtype: int64
```

```
##13. Which job title has the highest pay?
db.groupby(['job_title'])['salary'].max().head().sort_values(ascending=False)
```

```
job_title
AI Scientist                         1335000
Applied Machine Learning Scientist    423000
3D Computer Vision Researcher         400000
Applied Data Scientist                380000
Analytics Engineer                    205300
Name: salary, dtype: int64
```

```
##14 . Is freelancing is worth or not?
db.groupby(['employment_type'])['salary'].agg(['mean','max','min','sum'])
```

employment_type	mean	max	min	sum
CT	184000.000000	416000	29000	920000
FL	48000.000000	100000	12000	192000
FT	331124.622449	30400000	4000	194701278
PT	85476.000000	400000	8760	854760

```

```

```
### CASE - 1 PART A

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
##1 1.Write a program to display "Welcome " if a number entered by user is a multiple of five otherwise pri

number = int(input("Enter a number: "))

if number % 5 == 0:
    print("Welcome")
else:
    print("Bye")
```

```
Enter a number: 4
Bye
```

```
##2 2. Write a program to find the largest number out of three numbers excepted from user ?
num1 = int(input("enter the first number:"))
num2 = int(input("enter the second number:"))
num3 = int(input("enter the third number:"))
largest = max(num1, num2, num3)
print("the largest number is :",largest)
```

```
enter the first number:45
enter the second number:98
enter the third number:98
the largest number is : 98
```

```
 ##3. Write a program to accept a number from 1 to 7 and display the name of the day like 1 for sunday , 2
day = int(input("Enter a number from 1 to 7: "))

if day == 1:
    print("Sunday")
elif day == 2:
    print("Monday")
elif day == 3:
    print("Tuesday")
elif day == 4:
    print("Wednesday")
elif day == 5:
    print("Thursday")
elif day == 6:
    print("Friday")
elif day == 7:
    print("Saturday")
else:
    print("Invalid input. Please enter a number from 1 to 7.")


```

```
Enter a number from 1 to 7: 4
Wednesday
```

```
#4. 4. Accept any city from the user and display monument of that city  ?
city = input("Enter a city: ")

```

```
if city == "Delhi":
    print("Monument: Red Fort")
elif city == "Agra":
    print("Monument: Taj Mahal")
elif city == "Jaipur":
    print("Monument: Jal Mahal")
else:
    print("Monument not found for the given city.")
```

```
Enter a city: jaipur
Monument not found for the given city.
```

```
# 5. Write a program to accept two numbers and mathematical operators and perform
#operation accordingly
#Like:
#Enter First Number: 7
#Enter Second Number : 9
#Enter operator : + (you can use different operators as well)
#Your Answer is : 16
First_number = int(input('First_number: '))
Second_number = int(input('Second_number: '))
operator = input('Enter operator (+,-,*,/,%)')
if operator == '+':
  print(First_number+Second_number)
elif operator == '-':
  print(First_number-Second_number)
elif operator == '*':
  print(First_number*Second_number)
elif operator == '%':
  print(First_number%Second_number)
else:
  print('Invalid')


```

```
First_number: 20
Second_number: 10
Enter operator (+,-,*,/,%)+
Invalid
```

```
#6 Check if the input is Leap Year , write a function
#We add a Leap Day on February 29, almost every four years.
#The leap day is an extra, or intercalary day and we add it to the shortest month of the year,
#February.
#In the Gregorian calendar three criteria must be taken into account to identify leap years:
#1. The year can be evenly divided by 4, is a leap year, unless:
#2. The year can be evenly divided by 100, it is NOT a leap year, unless:
#3. The year is also evenly divisible by 400. Then it is a leap year.
#Examples : This means that in the Gregorian calendar, the years 2000 and 2400 are leap
#years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years.
#What you have to do?
#You are given the year, and you have to write a function to check if the year is leap or not.
#Note that you have to complete the function and remaining code is given as template.
#You can use a variable as input with a fix value of user input
Year = int(input('Enter your year here: '))
if Year%400==0:
  print('Leap Year')
elif Year%100==0:
  print('No Leap Year')
elif Year%4==0:
  print('Leap Year')
else:
  print('No Leap Year')
```

```
Enter your year here: 2220
Leap Year
```

```
#7 # Rock Paper Scissors , if else or while ?
#Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input),
#compare them, print out a message of congratulations to the winner, and ask if the players
#want to start a new game)
#Remember the rules:
#1. Rock beats scissors
#2. Scissors beats paper
#3. Paper beats rock
#If you don't know what is this game , go watch it :)

Player_1 = str(input())
Player_2 = str(input())
if Player_1==Player_2:
  print('Tie')
elif Player_1 == "Rock" and Player_2 == "Scissors":
  print('Player_1 wins')
elif Player_1 == "Rock" and Player_2 == "Paper":
  print('Player_2 wins')
elif Player_1 == 'Paper' and Player_2 == 'Rock':
  print('Player_2 wins')
elif Player_1 == "paper" and Player_2 == "Scissors":
  print('Player_2 wins')
elif Player_1 == "Scissors" and Player_2 == "Rock":
 print('Player_1 wins')
else:
 print('Player_1 wins')
```

```
Rock
paper
Player_1 wins
```

```
1
```