ASSIGNMENT NO: 4

GROUP: G-1

MEMBER NAME: MONALI BABDE (205)
ARYA GAJBHIYE (219)
AYUSH ABHANG (201)

```
import pandas as pd
df = pd.read_csv("/content/sample_data/salary.csv")

#print all records of dataset
print(df)
#print Education level of all employees
print(df['Education Level'])
#Print Education level and salaries of all employees
print(df[['Education Level','Salary']])
```

output:

```
    Age  Gender     Education Level          Job Title
0   32.0    Male       Bachelor's           Software Engineer
1   28.0  Female         Master's         Data Analyst
2   45.0    Male            PhD             Senior Manager
3   36.0  Female       Bachelor's       Sales Associate
4   52.0    Male         Master's           Director
...  ...   ...            ...              ...
6699  49.0  Female         PhD           Director of Marketing
6700  32.0    Male       High School       Sales Associate
6701  30.0  Female    Bachelor's Degree     Financial Manager
6702  46.0    Male    Master's Degree     Marketing Manager
6703  26.0  Female       High School       Sales Executive

    Years of Experience    Salary
0                   5.0   90000.0
1                   3.0   65000.0
2                  15.0  150000.0
3                   7.0   60000.0
4                  20.0  200000.0
...                 ...      ...
6699               20.0  200000.0
6700                3.0   50000.0
6701                4.0   55000.0
6702               14.0  140000.0
6703                1.0   35000.0

[6704 rows x 6 columns]
0         Bachelor's
1           Master's
2              PhD
3         Bachelor's
```

```
4            Master's
          ...
6699              PhD
6700       High School
6701   Bachelor's Degree
6702    Master's Degree
6703       High School
Name: Education Level, Length: 6704, dtype: object
     Education Level    Salary
0         Bachelor's   90000.0
1          Master's   65000.0
2               PhD  150000.0
3         Bachelor's   60000.0
4          Master's  200000.0
...              ...      ...
6699            PhD  200000.0
6700     High School   50000.0
6701  Bachelor's Degree   55000.0
6702  Master's Degree  140000.0
6703     High School   35000.0

[6704 rows x 2 columns]
```

```
# print salary and Gender
df1 = df[['Salary','Gender']]
print(df1)
```

```
     Salary   Gender
0    90000.0    Male
1    65000.0  Female
2   150000.0    Male
3    60000.0  Female
4   200000.0    Male
...      ...    ...
6699  200000.0  Female
6700  50000.0    Male
6701  55000.0  Female
6702  140000.0    Male
6703  35000.0  Female

[6704 rows x 2 columns]
```

```
# save DataFrame to a CSV file
df1.to_csv("Salary.csv",index=True)

# print all record through salary_data
salary_data = pd.read_csv('/content/sample_data/salary.csv')
salary_data
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0
6703	26.0	Female	High School	Sales Executive	1.0	35000.0

6704 rows × 6 columns

```
# compute basic summary statistics of salary_data
salary_data.describe()
```

	Age	Years of Experience	Salary
count	6702.000000	6701.000000	6699.000000
mean	33.620859	8.094687	115326.964771
std	7.614633	6.059003	52786.183911
min	21.000000	0.000000	350.000000
25%	28.000000	3.000000	70000.000000
50%	32.000000	7.000000	115000.000000
75%	38.000000	12.000000	160000.000000

	Age	Years of Experience	Salary
max	62.000000	34.000000	250000.000000

```
# To print the full summary of salary_data
salary_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6704 entries, 0 to 6703
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Age                  6702 non-null   float64
 1   Gender               6702 non-null   object 
 2   Education Level      6701 non-null   object 
 3   Job Title            6702 non-null   object 
 4   Years of Experience  6701 non-null   float64
 5   Salary               6699 non-null   float64
dtypes: float64(3), object(3)
memory usage: 314.4+ KB
```

```
# print Age
salary_data['Age'] =
salary_data['Age'].fillna(salary_data['Age'].mean())
salary_data['Age']
```

```
0    32.0
1    28.0
2    45.0
3    36.0
4    52.0
    ...
6699  49.0
6700  32.0
6701  30.0
6702  46.0
6703  26.0
Name: Age, Length: 6704, dtype: float64
```

```
salary_data['Years of Experience'].fillna(salary_data['Years of
Experience'].mean())
salary_data['Salary'] =
salary_data['Salary'].fillna(salary_data['Salary'].mean())
# replacing DataFrame from csv file
```

```
salary_data
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0
6703	26.0	Female	High School	Sales Executive	1.0	35000.0

6704 rows × 6 columns

```
import pandas as pd

df = pd.read_csv("/content/sample_data/salary.csv")
# print job title and year of experience
print(df[['Job Title','Years of Experience']])
```

```
                  Job Title  Years of Experience
0         Software Engineer                  5.0
1              Data Analyst                  3.0
2            Senior Manager                 15.0
3           Sales Associate                  7.0
4                  Director                 20.0
...                     ...                  ...
6699   Director of Marketing                 20.0
```

```
6700        Sales Associate                 3.0
6701      Financial Manager                 4.0
6702      Marketing Manager                14.0
6703        Sales Executive                 1.0

[6704 rows x 2 columns]
```

```
# compute basic summary statistics
print(df.describe())
```

	Age	Years of Experience	Salary
Count	6702.000000	6701.000000	6699.000000
mean	33.620859	8.094687	115326.964771
std	7.614633	6.059003	52786.183911
min	21.000000	0.000000	350.000000
25%	28.000000	3.000000	70000.000000
50%	32.000000	7.000000	115000.000000
75%	38.000000	12.000000	160000.000000
max	62.000000	34.000000	250000.000000

```
df.replace('Education Level','Gender')
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0

6703	26.0	Female	High School	Sales Executive	1.0	35000.0

6704 rows × 6 columns

```
# find the average values in each column
print(df.mean())
```

```
Age                         33.620859
Years of Experience          8.094687
Salary                  115326.964771
dtype: float64
```

```
# find the median values in each column
print(df.median())
```

```
Age                         32.0
Years of Experience          7.0
Salary                  115000.0
dtype: float64
```

```
# find the maximum values in each column
print(df.max())
```

```
Age                         62.0
Years of Experience         34.0
Salary                  250000.0
dtype: float64
```

```
# find the minimum values in each column
print(df.min())
```

```
Age                      21.0
Years of Experience       0.0
Salary                  350.0
dtype: float64
```

```
# find the sum of values in each column
df.sum()
```

```
Age                     225327.0
Years of Experience      54242.5
Salary                772575337.0
dtype: float64
```

```
# find the max value from Salary
print(df['Salary'].max())
```

```
250000.0
```

```
# find the max value from Age
print(df['Age'].max())
```

62.0

```
# find basic summary statistics of Salary
print(df['Salary'].describe())
```

```
count      6699.000000
mean     115326.964771
std       52786.183911
min         350.000000
25%       70000.000000
50%      115000.000000
75%      160000.000000
max      250000.000000
Name: Salary, dtype: float64
```

```
# count total number of non-missing values in each row
print(df.count())
```

```
Age                  6702
Gender               6702
Education Level      6701
Job Title            6702
Years of Experience  6701
Salary               6699
dtype : int64
```

```
# count the occurrences of each education level value in a column
print(df['Education Level'].value_counts())
```

```
Bachelor's Degree    2267
Master's Degree      1573
PhD                  1368
```

```
Bachelor's            756
High School           448
Master's              288
phD                     1
Name: Education Level, dtype: int64
```

```
# selecting salary >100000
print (df.loc[df['Salary']>100000])
```

```
Age  Gender    Education Level                         Job Title  \
2     45.0    Male                PhD                   Senior Manager
4     52.0    Male          Master's                          Director
6     42.0  Female          Master's                   Product Manager
9     38.0    Male                PhD                  Senior Scientist
11    48.0  Female         Bachelor's                       HR Manager
...    ...     ...                ...                              ...
6690  42.0    Male  Bachelor's Degree                Financial Manager
6693  43.0  Female    Master's Degree                    Sales Manager
6697  51.0  Female    Master's Degree  Senior Product Marketing Manager
6699  49.0  Female                PhD            Director of Marketing
6702  46.0    Male    Master's Degree                Marketing Manager

      Years of Experience    Salary
2                    15.0  150000.0
4                    20.0  200000.0
6                    12.0  120000.0
9                    10.0  110000.0
11                   18.0  140000.0
...                   ...       ...
6690                 13.0  130000.0
6693                 14.0  140000.0
6697                 19.0  190000.0
6699                 20.0  200000.0
6702                 14.0  140000.0

[3772 rows x 6 columns]
```

```
# Group by a Salary and compute the sum of Salary
df.groupby('Salary').sum()
```

Salary	Age	Years of Experience
350.0	29.0	1.5
500.0	31.0	4.0
550.0	25.0	1.0

Age	Years of Experience	
Salary		
579.0	23.0	1.0
25000.0	3296.0	42.0
...	...	...
220000.0	528.0	232.0
225000.0	400.0	184.0
228000.0	49.0	23.0
240000.0	408.0	192.0
250000.0	147.0	70.0

444 rows × 2 columns

```
# Group by Gender and Age and compute the mean for each group
df.groupby(['Gender', 'Age']).mean()
```

		Years of Experience	Salary
Gender	Age		
Female	**21.0**	0.000000	25000.000000
	22.0	0.000000	30722.000000
	23.0	0.846939	46174.530612
	24.0	0.791045	37552.888060
	25.0	1.562147	64330.790960
...	**...**	...	...
Other	**25.0**	2.000000	69032.000000
	31.0	8.000000	104127.000000
	37.0	14.000000	161393.000000
	53.0	31.000000	166109.000000
	54.0	29.000000	158788.000000

84 rows × 2 columns

```
# Apply multiple aggregation functions to Salary
df.groupby('Salary').agg(['mean', 'max', 'min'])
```

	Age			Years of Experience		
	mean	max	min	mean	max	min
Salary						
350.0	29.000000	29.0	29.0	1.500000	1.5	1.5
500.0	31.000000	31.0	31.0	4.000000	4.0	4.0
550.0	25.000000	25.0	25.0	1.000000	1.0	1.0
579.0	23.000000	23.0	23.0	1.000000	1.0	1.0
25000.0	24.781955	33.0	21.0	0.315789	1.0	0.0
...	...	...	...	...	...	...
220000.0	48.000000	49.0	44.0	21.090909	22.0	16.0
225000.0	50.000000	50.0	50.0	23.000000	23.0	23.0
228000.0	49.000000	49.0	49.0	23.000000	23.0	23.0
240000.0	51.000000	51.0	51.0	24.000000	24.0	24.0
250000.0	49.000000	52.0	45.0	23.333333	25.0	21.0

444 rows × 6 columns

```
# group by a column and count value from Age group
print(df.groupby('Age').count())
```

```
      Gender  Education Level  Job Title  Years of Experience  Salary
Age
21.0      18               18         18                   18      18
22.0      15               15         15                   15      15
23.0     104              104        104                  104     104
24.0     240              240        240                  240     240
25.0     284              284        284                  284     284
26.0     394              394        394                  393     393
27.0     517              516        517                  517     517
28.0     429              429        429                  429     429
29.0     444              444        444                  444     444
30.0     449              449        449                  449     449
31.0     365              365        365                  365     364
32.0     351              351        351                  351     351
```

```
33.0     398                398         398                    398      398
34.0     309                309         309                    309      309
35.0     200                200         200                    200      200
36.0     282                282         282                    282      281
37.0     156                156         156                    156      156
38.0     149                149         149                    149      149
39.0     158                158         158                    158      158
40.0      92                 92          92                     92       92
41.0     129                129         129                    129      129
42.0     176                176         176                    176      176
43.0     158                158         158                    158      158
44.0     126                126         126                    126      126
45.0     144                144         144                    144      144
46.0     102                102         102                    102      102
47.0      47                 47          47                     47       47
48.0      98                 98          98                     98       98
49.0      91                 91          91                     91       91
50.0      88                 88          88                     88       88
51.0      30                 30          30                     30       30
52.0      29                 29          29                     29       29
53.0       7                  7           7                      7        7
54.0      68                 68          68                     68       68
55.0      16                 16          16                     16       16
56.0      11                 11          11                     11       11
57.0       9                  9           9                      9        9
58.0       7                  7           7                      7        7
60.0       5                  5           5                      5        5
61.0       2                  2           2                      2        2
62.0       5                  5           5                      5        5
```

```
# group by column and compute the given value from salary column
print(df.groupby('Salary').get_group(250000))
```

```
Age Gender    Education Level                  Job Title  \
30    50.0   Male         Bachelor's                        CEO
83    52.0   Male                PhD  Chief Technology Officer
5001  45.0   Male  Bachelor's Degree         Financial Manager

      Years of Experience    Salary
30                   25.0  250000.0
83                   24.0  250000.0
5001                 21.0  250000.0
```

```
# group by a column and count value for each group of Gender
print(df.groupby('Gender').count())
```

```
Age  Education Level  Job Title  Years of Experience  Salary
Gender
Female  3014             3014       3014                 3013    3013
Male    3674             3673       3674                 3674    3672
Other     14               14         14                   14      14
```

```
# find the sum of values in year of experience
print(df['Years of Experience'].sum())
```

```
54242.5
```

```
# find the sum of values in salary
print(df['Salary'].sum())
```

```
772575337.0
```

```
# find the max value from year of experience
print(df['Years of Experience'].max())
```

```
34.0
```

```
# find the correlation between columns
print(df.corr())
```

```
                       Age  Years of Experience    Salary
Age                1.000000             0.937655  0.728053
Years of Experience  0.937655             1.000000  0.808969
Salary             0.728053             0.808969  1.000000
```

```
# find the covariance between columns
print(df.cov())
```

```
                              Age  Years of Experience        Salary
Age                     57.982630            43.260648  2.926778e+05
Years of Experience     43.260648            36.711518  2.587702e+05
Salary              292677.795581        258770.183028  2.786381e+09
```

```
df.isnull()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary	
0	False		False	False	False	False	False
1	False		False	False	False	False	False
2	False		False	False	False	False	False
3	False		False	False	False	False	False
4	False		False	False	False	False	False
...	...		...	...	...	...	...
6699	False		False	False	False	False	False
6700	False		False	False	False	False	False

	Age	Gender	Education Level	Job Title	Years of Experience	Salary	
6701	False	False	False	False	False	False	False
6702	False	False	False	False	False	False	False
6703	False	False	False	False	False	False	False

6704 rows × 6 columns

```
# Drops rows with any missing value
df.dropna()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary	
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0	
1	28.0	Female	Master's	Data Analyst	3.0	65000.0	
2	45.0	Male	PhD	Senior Manager	15.0	150000.0	
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0	
4	52.0	Male	Master's	Director	20.0	200000.0	
...	...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0	
6700	32.0	Male	High School	Sales Associate	3.0	50000.0	
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0	
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0	
6703	26.0	Female	High School	Sales Executive	1.0	35000.0	

6698 rows × 6 columns

```
# Drops columns with any missing value
df.dropna(axis=1)
```

0
1
2

3	
4	
...	
6699	
6700	
6701	
6702	
6703	

6704 rows × 0 columns

```
# Fill missing value with a specific value
df.fillna('Age')
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0
6703	26.0	Female	High School	Sales Executive	1.0	35000.0

6704 rows × 6 columns

```
# To check for duplicate rows in a DataFrame:
df.duplicated()
```

```
0     False
1     False
2     False
3     False
4     False
      ...
6699   True
6700   True
6701   True
6702   True
6703   True
Length: 6704, dtype: bool
```

```
# To drop duplicate rows:
df.drop_duplicates()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6623	43.0	Female	Master's Degree	Digital Marketing Manager	15.0	150000.0
6624	27.0	Male	High School	Sales Manager	2.0	40000.0
6625	33.0	Female	Bachelor's Degree	Director of Marketing	8.0	80000.0
6628	37.0	Male	Bachelor's Degree	Sales Director	7.0	90000.0
6631	30.0	Female	Bachelor's Degree	Sales Manager	5.0	70000.0

1792 rows × 6 columns

```
df.loc[df['Job Title'] == 'software Engineer', 'Job Title'] = 'Data
Analyst'
print(df)
```

```
Age  Gender    Education Level        Job Title  Years of Experience  \
0    32.0    Male       Bachelor's       bachelor's             5.0
1    28.0  Female         Master's         master's             3.0
2    45.0    Male            PhD              phd            15.0
3    36.0  Female       Bachelor's       bachelor's             7.0
4    52.0    Male         Master's         master's            20.0
...   ...     ...              ...              ...             ...
6699 49.0  Female            PhD              phd            20.0
6700 32.0    Male      High School      high school             3.0
6701 30.0  Female  Bachelor's Degree  bachelor's degree             4.0
6702 46.0    Male    Master's Degree    master's degree            14.0
6703 26.0  Female       High School      high school             1.0

      Salary
0    90000.0
1    65000.0
2   150000.0
3    60000.0
4   200000.0
...      ...
6699 200000.0
6700  50000.0
6701  55000.0
6702 140000.0
6703  35000.0

[6704 rows x 6 columns]
```

```
df['Job Title'] = df['Job Title'].str.strip()
print(df)
```

```
Age  Gender    Education Level          Job Title  Years of Experience
\
0     32.0    Male         Bachelor's         bachelor's
5.0
1     28.0  Female           Master's           master's
3.0
2     45.0    Male                PhD                phd
15.0
3     36.0  Female         Bachelor's         bachelor's
7.0
4     52.0    Male           Master's           master's
20.0
...    ...     ...                ...                ...
...
6699  49.0  Female                PhD                phd
20.0
6700  32.0    Male        High School        high school
3.0
6701  30.0  Female  Bachelor's Degree  bachelor's degree
4.0
```

```
6702  46.0    Male    Master's Degree    master's degree
14.0
6703  26.0  Female        High School        high school
1.0

        Salary
0      90000.0
1      65000.0
2     150000.0
3      60000.0
4     200000.0
...        ...
6699   200000.0
6700   50000.0
6701   55000.0
6702   140000.0
6703   35000.0

[6704 rows x 6 columns]
```

```
# Create two DataFrames
df1 = pd.DataFrame({'Age': [1, 2, 3],
                    'Job Title': ['Software Engineer  ', 'Sales
Manager', 'Data Analyst']})
df2 = pd.DataFrame({'Age': [4, 5, 6],
                    'Job Title': ['Senior Manager', 'Digital Marketing
Manager', 'Director of Marketing']})

# Concatenate the DataFrames vertically
concatenated_df = pd.concat([df1, df2], axis=0)

print(concatenated_df)

print(df)
```

```
Age                  Job Title
0    1        Software Engineer\t
1    2              Sales Manager
2    3               Data Analyst
0    4             Senior Manager
1    5   Digital Marketing Manager
2    6      Director of Marketing
       Age   Gender    Education Level              Job Title
0     32.0    Male         Bachelor's      Software Engineer
1     28.0   Female           Master's           Data Analyst
2     45.0    Male                PhD         Senior Manager
3     36.0   Female         Bachelor's        Sales Associate
4     52.0    Male           Master's                Director
...    ...     ...                ...                    ...
6699   49.0   Female                PhD   Director of Marketing
6700   32.0    Male        High School        Sales Associate
6701   30.0   Female   Bachelor's Degree      Financial Manager
6702   46.0    Male    Master's Degree      Marketing Manager
6703   26.0   Female        High School        Sales Executive
```

```
      Years of Experience    Salary
0                     5.0   90000.0
1                     3.0   65000.0
2                    15.0   150000.0
3                     7.0   60000.0
4                    20.0   200000.0
...                   ...       ...
6699                 20.0   200000.0
6700                  3.0   50000.0
6701                  4.0   55000.0
6702                 14.0   140000.0
6703                  1.0   35000.0

[6704 rows x 6 columns]
```

```
import pandas as pd

# Create two DataFrames
df1 = pd.DataFrame({"Years of Experience": [5.0, 20.0, 3.0],
                    "Education Level": ["Master's", "High School", "
Bachelor's"]})

df2 = pd.DataFrame({"Years of Experience": [14.0, 4.0, 1.0],
                    "Education Level": [" Bachelor's Degree" , " High
School ", "Master's"]})

# Concatenate the DataFrames vertically
concatenated_df = pd.concat([df1, df2], axis=0)

print(concatenated_df)
```

```
  Years of Experience     Education Level
0                   5.0           Master's
1                  20.0         High School
2                   3.0          Bachelor's
0                  14.0   Bachelor's Degree
1                   4.0        High School
2                   1.0           Master's
```

```
# Create a DataFrame

data = {
      "Years of Experience": [14.0, 4.0, 1.0],
        "Age": [4, 5, 6]
    }

df = pd.DataFrame(data)
print(df)
```

```
# Perform stack operation

stacked_df = df.set_index("Years of Experience").stack()
print(stacked_df)

# Perform unstack operation
unstacked_df = stacked_df.unstack()
print(unstacked_df)
```

```
             Years of Experience        Age
0                           14.0          4
1                            4.0          5
2                            1.0          6
Years of Experience
14.0                  Age    4
4.0                   Age    5
1.0                   Age    6
dtype: int64
                      Age
Years of Experience
14.0                    4
4.0                     5
1.0                     6
```

```
# Create a DataFrame
data = {
    "Years of Experience": [14.0, 4.0, 1.0,20.0],
      "Job Title ": ["Marketing Manager", "Senior Manager", "Digital
Marketing Manager", "Director of Marketing"],
      "Age": [4, 5, 6,9]
      }
df = pd.DataFrame (data)
print (df)
```

```
   Years of Experience                  Job Title   Age
0                 14.0          Marketing Manager     4
1                  4.0             Senior Manager     5
2                  1.0    Digital Marketing Manager     6
3                 20.0       Director of Marketing     9
```

```
df = pd.DataFrame({
         "Years of Experience": [14.0, 4.0, 1.0],
        "Age": [4, 5, 6]})

melted_df = df.melt(id_vars="Years of Experience", var_name="Age")
print(melted_df)
```

	Years of Experience	Age	value
0	14.0	Age	4
1	4.0	Age	5
2	1.0	Age	6

```
# selecting salary >100000
print(df.loc[df['Salary']>100000])
```

```
       Age  Gender    Education Level                         Job Title
\
2     45.0    Male                PhD                    Senior Manager
4     52.0    Male           Master's                           Director
6     42.0  Female           Master's                    Product
Manager
9     38.0    Male                PhD                  Senior Scientist
11    48.0  Female         Bachelor's                           HR
Manager
...    ...     ...                ...                                 ...
6690  42.0    Male  Bachelor's Degree                  Financial
Manager
6693  43.0  Female    Master's Degree                      Sales
Manager
6697  51.0  Female    Master's Degree  Senior Product Marketing
Manager
6699  49.0  Female                PhD             Director of
Marketing
6702  46.0    Male    Master's Degree                  Marketing
Manager

      Years of Experience    Salary
2                    15.0  150000.0
4                    20.0  200000.0
6                    12.0  120000.0
9                    10.0  110000.0
11                   18.0  140000.0
...                   ...       ...
6690                 13.0  130000.0
6693                 14.0  140000.0
6697                 19.0  190000.0
6699                 20.0  200000.0
6702                 14.0  140000.0

[3772 rows x 6 columns]
```

```
# selecting all rows and column from -1
print(df.iloc[::-1])
```

	Years of Experience	Age
2	1.0	6
1	4.0	5
0	14.0	4

```
df.fillna(0)
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0
6703	26.0	Female	High School	Sales Executive	1.0	35000.0

6704 rows × 6 columns

```
df.dropna()
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...	...	...	...	...	...	...
6699	49.0	Female	PhD	Director of Marketing	20.0	200000.0
6700	32.0	Male	High School	Sales Associate	3.0	50000.0

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
6701	30.0	Female	Bachelor's Degree	Financial Manager	4.0	55000.0
6702	46.0	Male	Master's Degree	Marketing Manager	14.0	140000.0
6703	26.0	Female	High School	Sales Executive	1.0	35000.0

6698 rows × 6 columns