LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CC5067NI-Smart Data Discovery

60% Individual Coursework

2023-24 Autumn

Student Name: Kirtan Maharjan

London Met ID: 22068180

College ID: np01cp4a220166@islingtoncollege.edu.np

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Monday, May 13, 2024

Word Count: 1867

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Contents

TABLE OF TABLES 2
1.Data Understanding: 3
2.Data Preparation 5
2.1.Write a python program to load data into pandas DataFrame 5
2.2.Write a python program to remove unnecessary columns i.e., salary and salary currency. 6
2.3.Write a python program to remove the NaN missing values from updated dataframe. 6
2.4.Write a python program to check duplicates value in the dataframe. 7
2.5.Write a python program to see the unique values from all the columns in the dataframe. 8
2.6.Rename the experience level columns as below. 9
3.Data Analysis 12
3.1.Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable. 12
3.2.Write a Python program to calculate and show correlation of all variables. 12
4.Data Exploration 14
4.1.Write a python program to find out top 15 jobs. Make a bar graph of sales as well. 14
4.2.Which job has the highest salaries? Illustrate with bar graph. 15
4.3.Write a python program to find out salaries based on experience level. Illustrate it through bar graph. 16
4.4.Write a Python program to show histogram and box plot of any chosen different variables. Use proper labels in the graph. 17

TABLE OF FIGURES

Figure 1:python program to load data into pandas DataFrame..... 5
Figure 2:python program to remove unnecessary columns i.e., salary and salary currency. 6
Figure 3:python program to remove the NaN missing values from updated dataframe. 6
Figure 4:a python program to check duplicates value in the dataframe. 7
Figure 5:python program to see the unique values from all the columns in the dataframe(i) 8
Figure 6:python program to see the unique values from all the columns in the dataframe(ii) 8
Figure 7:SE – Senior Level/Expert 9
Figure 8:MI – Medium Level/Intermediate 9
Figure 9:EN – Entry Level 10
Figure 10:EX – Executive Level 11
Figure 11:Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable. 12
Figure 12:Python program to calculate and show correlation of all variables. 12
Figure 13:python program to find out top 15 jobs and a bar graph of sales as well.(i) 14
Figure 14python program to find out top 15 jobs and a bar graph of sales as well.(ii) 14
Figure 15:Python program to find the highest salaries 15
Figure 16:python program to find out salaries based on experience level 16
Figure 17:Python program to show histogram and box plot of any chosen different variables(i) 17
Figure 18:Python program to show histogram and box plot of any chosen different variables(ii) 17

TABLE OF TABLES

Table 1 : Data Understanding 4

1.Data Understanding:

The provided data'Data Science.csv' appears to be a collection of information about science professionals. It includes details such as their experience level, employment type(full-time,contract), job title,salary,and residence location. Suprisingly, it aslo includes the salary information in both the orginal currency and converted to USD, allowing it for cross-country comparisons. It also has the details about the work environment like remote work percentage and company characteristics like size and location are present.

S.no	Column Name	Description	Data Type
1.	Work_year	Categorical variable representing the year of the work.	int
2.	Experience_level	Categorical variable representing experience level(SE,MI,EN,EX)	String
3.	Employment_type	Categorical variable representing employment type(FT,CT)	String
4.	Job_title	Categorical variable representing job title(Data Scientist,ML Engineer, etc.)	String
5.	salary	Numerical value representing salary	Int/float
6.	Salary_currency	Categorical variable representing salary currency(USD,EUR,INR)	String
7.	Salary_in_usd	Numerical value representing salary converted into USD	Int/Float
8.	employee_residence	Categorical variable representing employee residence country(US,ES,IN,etc.)	String
9	Remote_ratio	Numerical value representing the percentage oof remotework	int
10.	Company_location	Categorical variable representing company location country (US,IN,ES,etc.)	String

11.	Company_size	Categorical variable representing company size(S,M,L)	String

Table 1 : Data Understanding

2.Data Preparation

2.1.Write a python program to load data into pandas DataFrame

```
In [1]: import pandas as pd                    #Importing pandas

        data = pd.read_csv("Data_Science.csv") #Reading the CSV file "Data_Science.csv" into a pandas DataFrame
        df = pd.DataFrame(data)                #Assigning the DataFrame to a variable named 'df'
        df                                     #Printing the DataFrame 'df'
```

Out[1]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	c
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100	ES	
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	100	US	
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	100	US	
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	100	CA	
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA	100	CA	
...	...	...	...	...	...	...	...	...	...	...	
3750	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	
3751	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US	
3752	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US	
3753	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US	
3754	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	

3755 rows × 11 columns

Figure 1:python program to load data into pandas DataFrame

Firstly, it imports pandas as pd. Then the code reads data from file named ”Data_Science.csv” and stores it in a pandas DataFrame named data.

Secondly, df = pd.DataFrame(data), reassigns the data to a new data frame named df.

Then finally, it prints the contents of the dataframe named df.

2.2.Write a python program to remove unnecessary columns i.e., salary and salary currency.

```
In [2]: Columns_remove = ['salary','salary_currency']# List of columns to remove

        df = df.drop(columns=Columns_remove)# Dropping the specified columns from the DataFrame 'df'
        df# Printing the DataFrame 'df' after removing columns
```

Out[2]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	SE	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	US	100	US	S
3	2023	SE	FT	Data Scientist	175000	CA	100	CA	M
4	2023	SE	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	US	100	US	L
3751	2021	MI	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	EN	FT	Data Scientist	105000	US	100	US	S
3753	2020	EN	CT	Business Data Analyst	100000	US	100	US	L
3754	2021	SE	FT	Data Science Manager	94665	IN	50	IN	L

3755 rows × 9 columns

Figure 2:python program to remove unnecessary columns i.e., salary and salary currency.

Here the code removes two specific columns “salary” and “salary_in_usd”.

Firstly we create a list named Columns_remove containing the names of the two specific columns.

Then we use drop method on DataFrame df.Finally,we print the modified DataFrame df again.

2.3.Write a python program to remove the NaN missing values from updated dataframe.

```
In [3]: df = df.dropna()# Removing rows with missing values (NaN) from DataFrame 'df'
        df# Printing the DataFrame 'df' after removing rows with NaN values
```

Out[3]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	SE	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	US	100	US	S
3	2023	SE	FT	Data Scientist	175000	CA	100	CA	M
4	2023	SE	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	US	100	US	L
3751	2021	MI	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	EN	FT	Data Scientist	105000	US	100	US	S
3753	2020	EN	CT	Business Data Analyst	100000	US	100	US	L

Figure 3:python program to remove the NaN missing values from updated dataframe.

Here we remove the NaN missing values from updated dataframe using .dropna().

It removes rows from the DataFrame df that contain mising values. It basically removes rows where at least one element is missing.

2.4.Write a python program to check duplicates value in the dataframe.

```
In [4]: Duplicate_Values = df[df.duplicated()]# Finding duplicate rows in DataFrame 'df'
        Duplicate_Values# Printing the DataFrame containing duplicate rows
```

Out[4]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
115	2023	SE	FT	Data Scientist	150000	US	0	US	M
123	2023	SE	FT	Analytics Engineer	289800	US	0	US	M
153	2023	MI	FT	Data Engineer	100000	US	100	US	M
154	2023	MI	FT	Data Engineer	70000	US	100	US	M
160	2023	SE	FT	Data Engineer	115000	US	0	US	M
...	...	...	...	...	...	...	...	...	...
3439	2022	MI	FT	Data Scientist	78000	US	100	US	M
3440	2022	SE	FT	Data Engineer	135000	US	100	US	M
3441	2022	SE	FT	Data Engineer	115000	US	100	US	M
3586	2021	MI	FT	Data Engineer	200000	US	100	US	L
3709	2021	MI	FT	Data Scientist	90734	DE	50	DE	L

Figure 4:a python program to check duplicates value in the dataframe.

Here, to check duplicates value in the dataframe we created new DataFrame named Duplicate_Values. The.duplicated() method returns rows in df that are duplicates of other rows. By default, it looks at all columns in the DataFrame to find duplicates. So, a row is deemed a duplicate if all of its values are identical to another row. The boolean series returned by.duplicated() is used to filter df and pick only True rows (duplicates). These duplicates are then assigned to a new DataFrame, Duplicate_Values.

2.5.Write a python program to see the unique values from all the columns in the dataframe.

```
In [5]: Unique_Values = {}# Dictionary to store unique values for each column
        for column in df.columns:# Looping through each column in the DataFrame
            Unique_Values[column] = df[column].unique()  # Finding unique values in the current column

        for column, values in Unique_Values.items():# Adding unique values for the column to the dictionary
            print(f"Unique values in '{column}':")# Printing the unique values for each column
            print(values)
            print()
```

```
Unique values in 'work_year':
[2023 2022 2020 2021]

Unique values in 'experience_level':
['SE' 'MI' 'EN' 'EX']

Unique values in 'employment_type':
['FT' 'CT' 'FL' 'PT']

Unique values in 'job_title':
['Principal Data Scientist' 'ML Engineer' 'Data Scientist'
 'Applied Scientist' 'Data Analyst' 'Data Modeler' 'Research Engineer'
 'Analytics Engineer' 'Business Intelligence Engineer'
 'Machine Learning Engineer' 'Data Strategist' 'Data Engineer'
 'Computer Vision Engineer' 'Data Quality Analyst'
 'Compliance Data Analyst' 'Data Architect'
 'Applied Machine Learning Engineer' 'AI Developer' 'Research Scientist'
 'Data Analytics Manager' 'Business Data Analyst' 'Applied Data Scientist'
 'Staff Data Analyst' 'ETL Engineer' 'Data DevOps Engineer' 'Head of Data'
```

Figure 5:python program to see the unique values from all the columns in the dataframe(i)

```
In [5]: Unique_Values = {}# Dictionary to store unique values for each column
        for column in df.columns:# Looping through each column in the DataFrame
            Unique_Values[column] = df[column].unique()  # Finding unique values in the current column

        for column, values in Unique_Values.items():# Adding unique values for the column to the dictionary
            print(f"Unique values in '{column}':")# Printing the unique values for each column
            print(values)
            print()
```

```
 'HR' 'PL' 'KW' 'VN' 'CY' 'AR' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK'
 'IT' 'MA' 'LT' 'BE' 'AS' 'IR' 'HU' 'SK' 'CN' 'CZ' 'CR' 'TR' 'CL' 'PR'
 'DK' 'BO' 'PH' 'DO' 'EG' 'ID' 'AE' 'MY' 'JP' 'EE' 'HN' 'TN' 'RU' 'DZ'
 'IQ' 'BG' 'JE' 'RS' 'NZ' 'MD' 'LU' 'MT']

Unique values in 'remote_ratio':
[100   0  50]

Unique values in 'company_location':
['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'NL' 'CH' 'CF' 'FR' 'FI' 'UA'
 'IE' 'IL' 'GH' 'CO' 'SG' 'AU' 'SE' 'SI' 'MX' 'BR' 'PT' 'RU' 'TH' 'HR'
 'VN' 'EE' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK' 'IT' 'MA' 'PL' 'AL'
 'AR' 'LT' 'AS' 'CR' 'IR' 'BS' 'HU' 'AT' 'SK' 'CZ' 'TR' 'PR' 'DK' 'BO'
 'PH' 'BE' 'ID' 'EG' 'AE' 'LU' 'MY' 'HN' 'JP' 'DZ' 'IQ' 'CN' 'NZ' 'CL'
 'MD' 'MT']

Unique values in 'company_size':
['L' 'S' 'M']
```

Figure 6:python program to see the unique values from all the columns in the dataframe(ii)

Firstly , we start with an empty dictionary named unique_value.it will act like a container to store the unique values found in each column.

Then, we loop through every column name in the data frame. Inside the loop it finds the unique values present in that specific column and stores them in the Unique_values dictionary. Finally, for each columns it prints a message specifying the column name and then prints the unique values found in that column.

2.6.Rename the experience level columns as below.

2.6.1.SE – Senior Level/Expert

```
In [6]: df['experience_level'] = df['experience_level'].replace("SE","Senior Level")# Replacing values in 'experience_level' column
        df# Printing the DataFrame 'df' after replacing values
```

Out[6]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	Senior Level	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	US	100	US	S
3	2023	Senior Level	FT	Data Scientist	175000	CA	100	CA	M
4	2023	Senior Level	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	Senior Level	FT	Data Scientist	412000	US	100	US	L
3751	2021	MI	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	EN	FT	Data Scientist	105000	US	100	US	S
3753	2020	EN	CT	Business Data Analyst	100000	US	100	US	L

Figure 7:SE – Senior Level/Expert

Here, this code targets the 'experience_level' column of the DataFrame df. It uses .replace() method which acts like a search and replace function.

In this case it searches "SE" with in the 'experience_level' column. Whenever it finds "SE" it replaces it with "Senior Level".

2.6.2.MI – Medium Level/Intermediate

```
In [7]: df['experience_level'] = df['experience_level'].replace("MI","Medium Level")# Replacing values in 'experience_level' column
        df# Printing the DataFrame 'df' after replacing values
```

Out[7]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	Senior Level	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	Medium Level	CT	ML Engineer	30000	US	100	US	S
2	2023	Medium Level	CT	ML Engineer	25500	US	100	US	S
3	2023	Senior Level	FT	Data Scientist	175000	CA	100	CA	M
4	2023	Senior Level	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	Senior Level	FT	Data Scientist	412000	US	100	US	L
3751	2021	Medium Level	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	EN	FT	Data Scientist	105000	US	100	US	S
3753	2020	EN	CT	Business Data Analyst	100000	US	100	US	L

Figure 8:MI – Medium Level/Intermediate

Here, this code targets the ‘experience_level’ column of the DataFrame df. It uses .replace() method which acts like a search and replace function.

In this case it searches “MI” with in the ‘experience_level’ column. Whenever it finds “MI” it replaces it with “Medium Level”.

2.6.3.EN – Entry Level

```
In [8]: df['experience_level'] = df['experience_level'].replace("EN","Entry Level")# Replacing values in 'experience_level' column
        df# Printing the DataFrame 'df' after replacing values
```

Out[8]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	Senior Level	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	Medium Level	CT	ML Engineer	30000	US	100	US	S
2	2023	Medium Level	CT	ML Engineer	25500	US	100	US	S
3	2023	Senior Level	FT	Data Scientist	175000	CA	100	CA	M
4	2023	Senior Level	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	Senior Level	FT	Data Scientist	412000	US	100	US	L
3751	2021	Medium Level	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	Entry Level	FT	Data Scientist	105000	US	100	US	S
3753	2020	Entry Level	CT	Business Data Analyst	100000	US	100	US	L
3754	2021	Senior Level	FT	Data Science Manager	94665	IN	50	IN	L

3755 rows × 9 columns

Figure 9:EN – Entry Level

Here, this code targets the ‘experience_level’ column of the DataFrame df. It uses .replace() method which acts like a search and replace function.

In this case it searches “EN” with in the ‘experience_level’ column. Whenever it finds “EN” it replaces it with “Entry Level”.

2.6.4.EX – Executive Level

```
In [9]: df['experience_level'] = df['experience_level'].replace("EX","Execute Level")# Replacing values in 'experience_level' column

        Execute_Level = df[df['experience_level'] == 'Execute Level']# Filtering rows where 'experience_level' is "Executive Level"
        Execute_Level# Printing the DataFrame containing rows with "Executive Level" experience
```

Out[9]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
58	2023	Execute Level	FT	Data Scientist	258750	US	0	US	M
59	2023	Execute Level	FT	Data Scientist	185000	US	0	US	M
183	2020	Execute Level	FT	Staff Data Analyst	15000	NG	0	CA	M
203	2023	Execute Level	FT	Data Engineer	220000	US	0	US	M
204	2023	Execute Level	FT	Data Engineer	205000	US	0	US	M
...	...	...	...	...	...	...	...	...	...
3559	2020	Execute Level	FT	Data Engineer	79833	ES	50	ES	L
3643	2021	Execute Level	FT	Director of Data Science	141846	DE	0	DE	L
3660	2021	Execute Level	FT	Data Science Consultant	69741	FR	100	ES	S
3675	2021	Execute Level	CT	Principal Data Scientist	416000	US	100	US	S

Figure 10:EX – Executive Level

Here, this code targets the 'experience_level' column of the DataFrame df. It uses .replace() method which acts like a search and replace function.

In this case it searches "EX" with in the 'experience_level' column. Whenever it finds "EX" it replaces it with "Executive Level".

3.Data Analysis

3.1.Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

```
In [10]: # Summary statistics for 'salary_in_usd' column

         # Sum
         sum_data = df['salary_in_usd'].sum()

         # Mean
         mean_data = df['salary_in_usd'].mean()

         # Standard deviation
         std_dev_data = df['salary_in_usd'].std()

         # Skewness
         skewness_data = df['salary_in_usd'].skew()

         # Kurtosis
         kurtosis_data = df['salary_in_usd'].kurt()


         # Printing summary statistics
         print("Summary Statistics:")
         print("Sum:", sum_data)
         print("Mean:", mean_data)
         print("Standard Deviation:", std_dev_data)
         print("Skewness:", skewness_data)
         print("Kurtosis:", kurtosis_data)

         Summary Statistics:
         Sum: 516576814
         Mean: 137570.38988015978
         Standard Deviation: 63055.625278224084
         Skewness: 0.5364011659712974
         Kurtosis: 0.8340064594833612
```

Figure 11:Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

This code calculates and displays various summary Statics for ‘Salary_in_usd’ column of the DataFrame df. It uses various method to calculate sum,mean,standard deviation,skewness and kurtosis like .mean(),.sum(),.std(),.skew() and .kurt().

3.2.Write a Python program to calculate and show correlation of all variables.

```
In [11]: # Creating a DataFrame from the data
         df = pd.DataFrame(data)

         # Selecting 'salary_in_usd' and 'work_year' columns for correlation analysis
         correlation_matrix = df[['salary_in_usd', 'work_year']].corr()

         # Printing the correlation matrix
         print("Correlation Matrix:")
         print(correlation_matrix)

         Correlation Matrix:
                       salary_in_usd  work_year
         salary_in_usd        1.00000    0.22829
         work_year            0.22829    1.00000
```

Figure 12:Python program to calculate and show correlation of all variables.

Here, this code calculates a statistical measure called correlation coefficient between the values on these two columns i.e. ‘salary_in_usd’ and ‘work_year’. The result is stored in a DataFrame named correlation_matrix.

4.Data Exploration

4.1.Write a python program to find out top 15 jobs. Make a bar graph of sales as well.

```
In [13]: # Creating a DataFrame from the data
df = pd.DataFrame(data)

# Selecting the top 15 most frequent job titles
First_Fifteen_jobs = df['job_title'].value_counts().head(15).index.tolist()

# Printing the top 15 job titles
print("Top 15 Most Frequent Job Titles:")
print(First_Fifteen_jobs)

Top 15 Most Frequent Job Titles:
['Data Engineer', 'Data Scientist', 'Data Analyst', 'Machine Learning Engineer', 'Analytics Engineer', 'Data Architect', 'Resea
rch Scientist', 'Data Science Manager', 'Applied Scientist', 'Research Engineer', 'ML Engineer', 'Data Manager', 'Machine Learn
ing Scientist', 'Data Science Consultant', 'Data Analytics Manager']
```

Figure 13:python program to find out top 15 jobs and a bar graph of sales as well.(i)

```
In [17]: # Importing libraries
import numpy as np
import matplotlib.pyplot as plt

# Creating a DataFrame from the data
df = pd.DataFrame(data)

# Selecting the top 15 most frequent job titles
First_Fifteen_jobs = df['job_title'].value_counts().head(15).index.tolist()

# Data for the bar chart
y = First_Fifteen_jobs  # List of top 15 job titles
x = np.arange(1,16)  # X-axis positions for bars

# Create the bar chart
plt.bar(x, y)
```

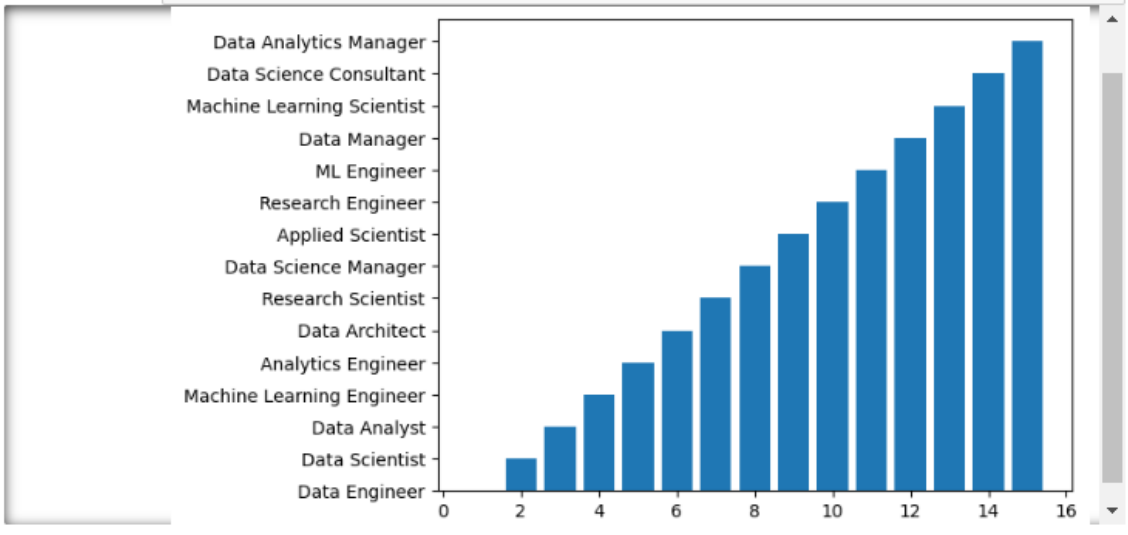

Figure 14python program to find out top 15 jobs and a bar graph of sales as well.(ii)

Here, we firstly import libraries ,the numpy library and Matplot library.

Then we create a pandas DataFrame as df. Then we find top fifteen jobs using .value_counts()(method to count the occurrences), ,head(15)(method to select the top 15 from this count) and .tolist()(converts the resulting index into a list).

Then finally, we create a bargraph.y is assigned the list of First_Fifteen_jobs and x is created using np.arange(1,16)(a Numpy array containing evenly spaced values from 1 to 15.)

4.2.Which job has the highest salaries? Illustrate with bar graph.

```
In [19]: # Finding the maximum salary in USD
highest_salary = df['salary_in_usd'].max()

# Filtering rows where salary equals the maximum salary
highest_salary_job_title = df[df['salary_in_usd'] == highest_salary]['job_title']

# Printing the job title with the highest salary
print("Job with Highest Salary:")
print(highest_salary_job_title)
```

```
Job with Highest Salary:
3522    Research Scientist
Name: job_title, dtype: object
```

Figure 15:Python program to find the highest salaries

Here we find the maximun salary using .max() salary.Then we filter(filters the DataFrame df to keep only rows where the salary in 'salary_in_usd' is equal to the highest_salary) and store the job title of the highest salary found by .max() in highest_salary_job_title. Then finally we print the highest salary's job title.

4.3.Write a python program to find out salaries based on experience level. Illustrate it through bar graph.

```
In [20]: # Grouping salaries by experience level and finding the maximum for each group
         exp_based_salary = df.groupby('experience_level')['salary_in_usd'].max()

         # Data for the bar chart
         x = df['experience_level']
         y = df['salary_in_usd']
         plt.bar(x, y)

Out[20]: <BarContainer object of 3755 artists>
```

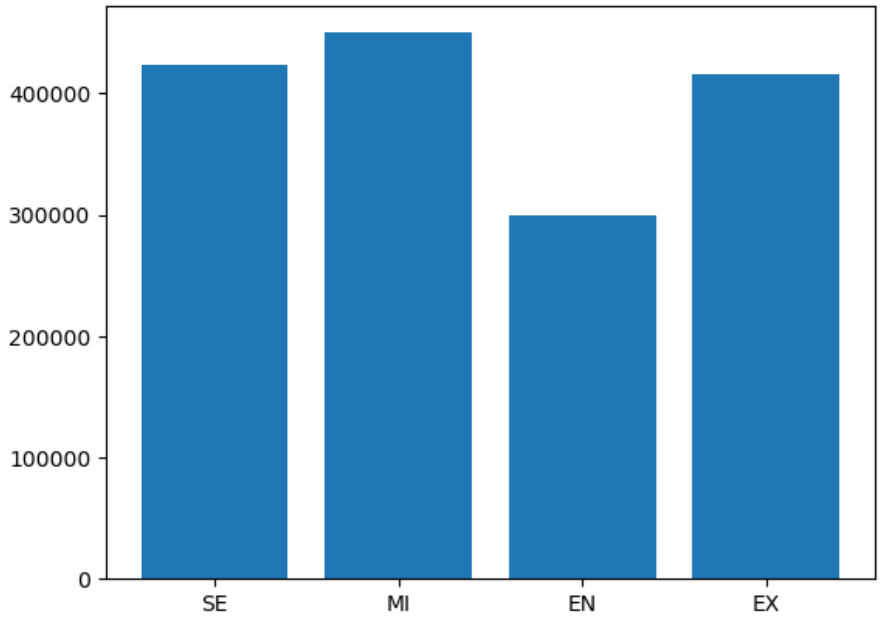

Figure 16:python program to find out salaries based on experience level

Firstly we fine the maximum salary per Experience Level using .max() method.

.groupby separates the data into different groups based on the experience level (Entry-Level, Mid-Level, Senior and executive).

Then finally we create a bar Chart where x assigned as Experience Level and y assigned as salaries in usd.

4.4.Write a Python program to show histogram and box plot of any chosen different variables. Use proper labels in the graph.

```
In [21]: # Configuring plot size
plt.figure(figsize=(10, 6))

# Creating a subplot grid of 1 row and 2 columns
plt.subplot(1, 2, 1)

# Creating a histogram of the salary_in_usd column with 20 bins and black edges for the bars
plt.hist(data['salary_in_usd'], bins=20, edgecolor='black')

# Adding a title to the subplot
plt.title('Histogram of Salary in USD')

# Labeling the x-axis (Salary Amount)
plt.xlabel('Salary Amount')

# Labeling the y-axis (Frequency)
plt.ylabel('Frequency')

# Moveing to the 2nd subplot on the right
plt.subplot(1, 2, 2)

# Creating a boxplot of the salary_in_usd column
plt.boxplot(data['salary_in_usd'])

# Adding a title to the subplot
plt.title('Box Plot of Salary in USD')

# Labeling the y-axis (Salary)
plt.ylabel('Salary')

# Displaying the plot
plt.show()
```

Figure 17:Python program to show histogram and box plot of any chosen different variables(i)

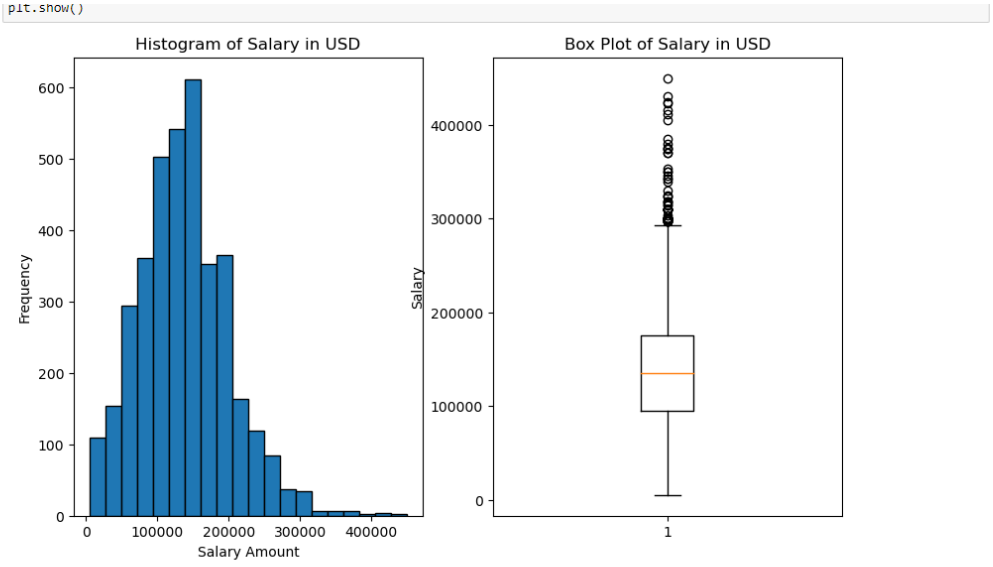

Figure 18:Python program to show histogram and box plot of any chosen different variables(ii)

Firstly we start by configuring the overall plot size to be 10 inches wide and 6 inches high using plt.figure(figsize=(10, 6)). This creates a canvas to hold the visualizations.

Then we creates a subplot grid with one row and two columns using plt.subplot(1, 2, 1). This essentially divides the canvas into two sections where each section will hold a separate plot.

Then we create a histogram for Salary in usd. Then we add context to the histogram.

Then we do the same for Boxplot.Finally we display the results.