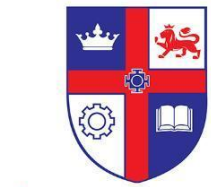
LONDON METROPOLITAN UNIVERSITY



Islington college
(इस्लिङ्टन कलेज)

CC5067NI-Smart Data Discovery

60% Individual Coursework

2023-24 Spring

Student Name: Sarthak Shrestha

London Met ID: 22067675

College ID: np01cp4a220100

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Sunday, May 12, 2024

Word Count: 947

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.

Acknowledgement

In this data science salary analysis coursework, I would like to thank our module teachers deeply for their invaluable tutorship and support. Their knowledge and professional advice helped a lot in determining the course of the project and, finally, in implementation of the desired ideas. Their openness to answering questions and giving clarification on the complicated topics was a key factor in my understanding of data analysis concepts and techniques.

Abstract

This report is the result of the analysis of a dataset that contains information about the variables that determine the data science salaries. The main goal was to get ready for analysis and discovery of salary patterns. Python programs were developed to achieve the following: data loading, cleaning, exploring and initial analysis. The report covers issues like the handling of missing values, data transformation, descriptive statistics, correlations, job distribution, salary variation by job title and experience level, and data visualization techniques like bar charts, histograms, and box plots. The data serves as a stepping stone, offering a first-hand knowledge of the data and laying the groundwork for further research on elements shaping the data science salaries.

Table of Contents

Table of Figures

Table of Tables

1. Data Understanding

The dataset is focused on data science professionals and their salaries. It contains information about various factors that could influence their salaries. The data set has various columns such as: work year, experience level, employment type, job title, salary, salary currency, salary in USD, employee residence, remote ratio, company location and company size. It appears that the dataset has the information of the work year 2020-2023. The different experience levels in this dataset are SE: Senior/Exert level, EX: Executive level, MI: Medium Level/Intermediate level and EN: Entry Level.

The columns in the dataset are described in the table below.

S.N.	Column Name	Description	Data Type
1	Work_year	This column contains the data of the work year of the employees.	int
2	Experience_level	This column contains the data of the different experience level of the employees.	str
3	Employment_type	This column contains the data of the different employment type of the employees.	str
4	Job_title	This column contains the data of the different job title of the employees.	str
5	Salary	This column contains the data of the salary of the employees.	Int
6	Salary_currency	This column contains the data of the different salary currency of the employees.	str
7	Salary_in_USD	This column contains the data of the salary in USD of the employees.	int

8	Employee_residence	This column contains the data of the different residence location of the employees.	str
9	Remote_ratio	This column contains the data of the different remote ratio of the employees.	int
10	Company_location	This column contains the data of the different company location of the employees.	str
11	Company_size	This column contains the data of the different company size of the employees.	str

Table 1: Description table

2. Data Preparation

2.1. Write a python program to load data into pandas Data Frame.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

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

```
In [2]: df=pd.read_csv('DataScienceSalaries.csv')
        df
```

Out[2]:

	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: Loading data in pandas data frame

First, we import the python libraries then we load the data into the pandas data frame and then displayed.

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

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

```
In [4]: df.pop("salary")
```

```
Out[4]: 0         80000
        1         30000
        2         25500
        3        175000
        4        120000
                 ...
        3750     412000
        3751     151000
        3752     105000
        3753     100000
        3754    7000000
        Name: salary, Length: 3755, dtype: int64
```

```
In [5]: df.pop("salary_currency")
```

```
Out[5]: 0       EUR
        1       USD
        2       USD
        3       USD
        4       USD
                ...
        3750    USD
        3751    USD
        3752    USD
        3753    USD
        3754    INR
        Name: salary_currency, Length: 3755, dtype: object
```

```
In [6]: df
```

Out[6]:

	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: Removing the unnecessary columns

Here, the unnecessary columns (salary and salary currency) are removed, and the updated data frame is displayed to show that the columns have been removed.

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

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

```
In [7]: df.isnull().sum()

Out[7]: work_year            0
        experience_level     0
        employment_type      0
        job_title            0
        salary_in_usd        0
        employee_residence   0
        remote_ratio         0
        company_location     0
        company_size         0
        dtype: int64
```

Figure 3: Checking for null values

First, we checked for null values, since there were no null values, we did not need to remove the NaN missing values from the updated data frame.

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

Making the data consistent

```
In [8]: df['job_title'] = df['job_title'].replace(
            ['Data Quality Analyst', 'Compliance Data Analyst',
             'Business Data Analyst', 'Staff Data Analyst',
             'Lead Data Analyst', 'Financial Data Analyst',
             'BI Data Analyst', 'Product Data Analyst',
             'Marketing Data Analyst', 'Principal Data Analyst',
             'Finance Data Analyst', 'Data Operations Analyst'], 'Data Analyst')

In [9]: df['job_title'] = df['job_title'].replace(
            ['Principal Data Scientist', 'Applied Data Scientist',
             'Lead Data Scientist', 'Data Scientist Lead',
             'Product Data Scientist', 'Staff Data Scientist',
             'Applied Scientist'], 'Data Scientist')

In [10]: df['job_title'] = df['job_title'].replace(
             ['Machine Learning Engineer', 'Applied Machine Learning Engineer',
              'Principal Machine Learning Engineer', 'Lead Machine Learning Engineer',
              'Machine Learning Infrastructure Engineer', 'Machine Learning Software Engineer',
              'Machine Learning Research Engineer', 'MLOps Engineer'], 'ML Engineer')

In [11]: df['job_title'] = df['job_title'].replace(
             ['Machine Learning Scientist', 'Applied Machine Learning Scientist'], 'Machine Learning Scientist')

In [12]: df['job_title'] = df['job_title'].replace('Big Data Engineer', 'Data Engineer')

In [13]: df['job_title'] = df['job_title'].replace(
             ['BI Data Engineer', 'Software Data Engineer',
              'Azure Data Engineer', 'Marketing Data Engineer',
              'Cloud Data Engineer', 'Lead Data Engineer',
              'Principal Data Engineer', 'Data Science Engineer',
              'Data Analytics Engineer', 'Data Infrastructure Engineer ',
              'Data Operations Engineer'], 'Data Engineer')

In [14]: df['job_title'] = df['job_title'].replace('Data Science Tech Lead', 'Data Science Lead')

In [15]: df['job_title'] = df['job_title'].replace('Head of Data Science', 'Head of Data')

In [16]: df['job_title'] = df['job_title'].replace(
             ['Data Management Specialist', 'Data Analytics Specialist'], 'Data Specialist')

In [17]: df['job_title'] = df['job_title'].replace(
             ['Data Science Manager', 'Data Analytics Manager', 'Manager Data Management'], 'Data Manager')

In [18]: df['job_title'] = df['job_title'].replace(
             ['Big Data Architect', 'Principal Data Architect', 'Cloud Data Architect'], 'Data Architect')

In [19]: df['job_title'] = df['job_title'].replace('Computer Vision Software Engineer', 'Computer Vision Engineer')

In [20]: df['job_title'] = df['job_title'].replace('Data Analytics Lead', 'Data Lead')
```

Figure 4: Making the data consistent

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

```
In [21]: df[df.duplicated()]
```

Out[21]:

	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
...	...	...	...	...	...	...	...	...	...
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
3616	2021	MI	FT	Data Engineer	110000	US	100	US	L
3709	2021	MI	FT	Data Scientist	90734	DE	50	DE	L

1197 rows × 9 columns

Figure 5: Checking for duplicate values

First, we made the data consistent, then we checked for the duplicate values which are then displayed.

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

```
In [39]: for column in df.columns:
             print(f"Unique values in '{column}':")
             print(df[column].unique())
```

```
Unique values in 'work_year':
[2023 2022 2020 2021]
Unique values in 'experience_level':
['Senior Level/Expert' 'Medium Level/Intermediate' 'Entry Level'
 'Executive Level']
Unique values in 'employment_type':
['FT' 'CT' 'FL' 'PT']
Unique values in 'job_title':
['Data Scientist' 'ML Engineer' 'Data Analyst' 'Data Modeler'
 'Research Engineer' 'Analytics Engineer' 'Business Intelligence Engineer'
 'Data Strategist' 'Data Engineer' 'Computer Vision Engineer'
 'Data Architect' 'AI Developer' 'Research Scientist' 'Data Manager'
 'ETL Engineer' 'Data DevOps Engineer' 'Head of Data'
 'Machine Learning Researcher' 'Data Specialist'
 'Director of Data Science' 'Machine Learning Scientist' 'AI Scientist'
 'Autonomous Vehicle Technician' 'Cloud Database Engineer'
 'Data Infrastructure Engineer' 'AI Programmer' 'BI Developer'
 'Data Science Lead' 'Deep Learning Researcher' 'BI Analyst'
 'Data Science Consultant' 'Insight Analyst' 'Deep Learning Engineer'
 'Data Lead' 'NLP Engineer' 'Machine Learning Developer'
 '3D Computer Vision Researcher' 'Data Analytics Consultant'
 'Power BI Developer' 'Machine Learning Manager' 'ETL Developer'
 'Head of Machine Learning']
Unique values in 'salary_in_usd':
[ 85847  30000  25500 ...  28369 412000  94665]
Unique values in 'employee_residence':
['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'PT' 'NL' 'CH' 'CF' 'FR' 'AU'
 'FI' 'UA' 'IE' 'IL' 'GH' 'AT' 'CO' 'SG' 'SE' 'SI' 'MX' 'UZ' 'BR' 'TH'
 '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: Displaying the unique values

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

```
In [22]: df.nunique()

Out[22]: work_year              4
         experience_level       4
         employment_type        4
         job_title             42
         salary_in_usd       1035
         employee_residence    78
         remote_ratio           3
         company_location      72
         company_size           3
         dtype: int64
```

Figure 7: Finding the number of unique values in each column

First, we checked the number of unique values in each column, then we displayed all the unique values in the column.

2.6. Rename the experience level columns as below

2.6. Rename the experience level columns as below.

SE – Senior Level/Expert, MI – Medium Level/Intermediate, EN – Entry Level, EX – Executive Level

```
In [23]: df['experience_level'].replace({'SE':'Senior Level/Expert',
                                         'EN':'Entry Level',
                                         'MI':'Medium Level/Intermediate',
                                         'EX':'Executive Level'}, inplace = True)
         df
```

Out[23]:

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

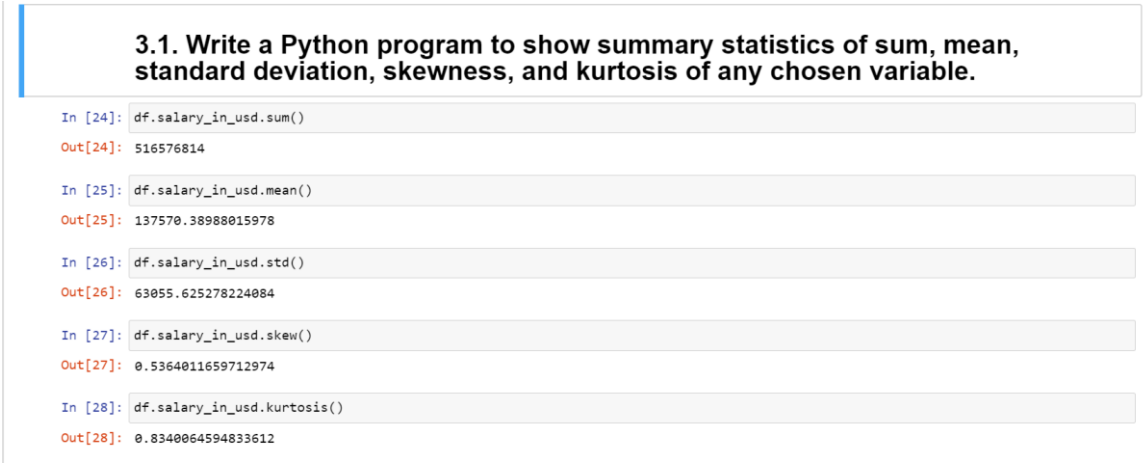
3755 rows × 9 columns

Figure 8: Renaming the experience levels

Here, we renamed the values in the experience level columns. Then we displayed the data frame to see if the values have been renamed.

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.



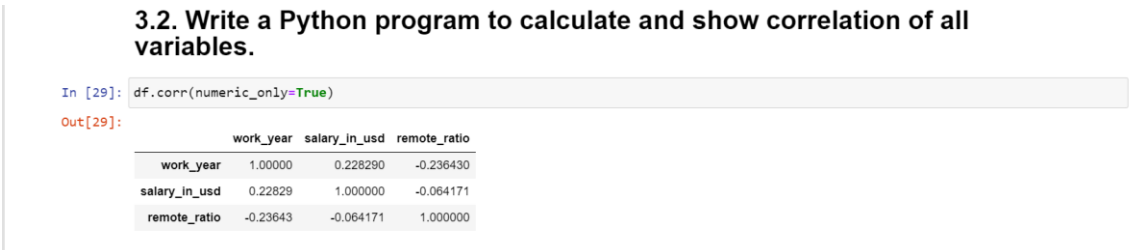
3.1. Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

```
In [24]: df.salary_in_usd.sum()
Out[24]: 516576814

In [25]: df.salary_in_usd.mean()
Out[25]: 137570.38988015978

In [26]: df.salary_in_usd.std()
Out[26]: 63055.625278224084

In [27]: df.salary_in_usd.skew()
Out[27]: 0.5364011659712974

In [28]: df.salary_in_usd.kurtosis()
Out[28]: 0.8340064594833612
```

Figure 9: summary statistics of sum, mean, standard deviation, skewness, and kurtosis of salary in USD

Here, we have calculated to sum, mean, standard deviation, skewness, and kurtosis of salary in USD and displayed the output.

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



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

```
In [29]: df.corr(numeric_only=True)
```

Out[29]:

	work_year	salary_in_usd	remote_ratio
work_year	1.00000	0.228290	-0.236430
salary_in_usd	0.22829	1.000000	-0.064171
remote_ratio	-0.23643	-0.064171	1.000000

Figure 10: Calculating and showing the correlation of all the numeric variables

Here, we have calculated and displayed the correlation of all the numeric variables.

4. Data Exploration

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

dropping the dublicate values

In [30]:
```
cdf = df.drop_duplicates()
cdf
```

Out[30]:

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

2558 rows × 9 columns

In [31]:
```
cdf[cdf.duplicated()]
```

Out[31]:

work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size

Figure 11: Dropping the duplicate values

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

In [41]:
```
job_title_counts = cdf['job_title'].value_counts()
top_15_jobs = job_title_counts.sort_values(ascending=False).head(15)
top_15_jobs
```

Out[41]:
```
job_title
Data Engineer                637
Data Scientist               593
Data Analyst                 454
ML Engineer                  263
Data Manager                  93
Analytics Engineer            91
Data Architect                68
Research Scientist            65
Machine Learning Scientist    38
Research Engineer             33
Computer Vision Engineer      23
Data Science Consultant       23
Head of Data                  19
AI Scientist                  16
Data Specialist               15
Name: count, dtype: int64
```

Figure 12: Finding the top 15 jobs

```
In [44]: plt.figure(figsize=(10, 6))
         plt.title('Top 15 Jobs')
         plt.ylabel('Frequency')
         top_15_jobs.plot(kind='bar')

Out[44]: <Axes: title={'center': 'Top 15 Jobs'}, xlabel='job_title', ylabel='Frequency'>
```

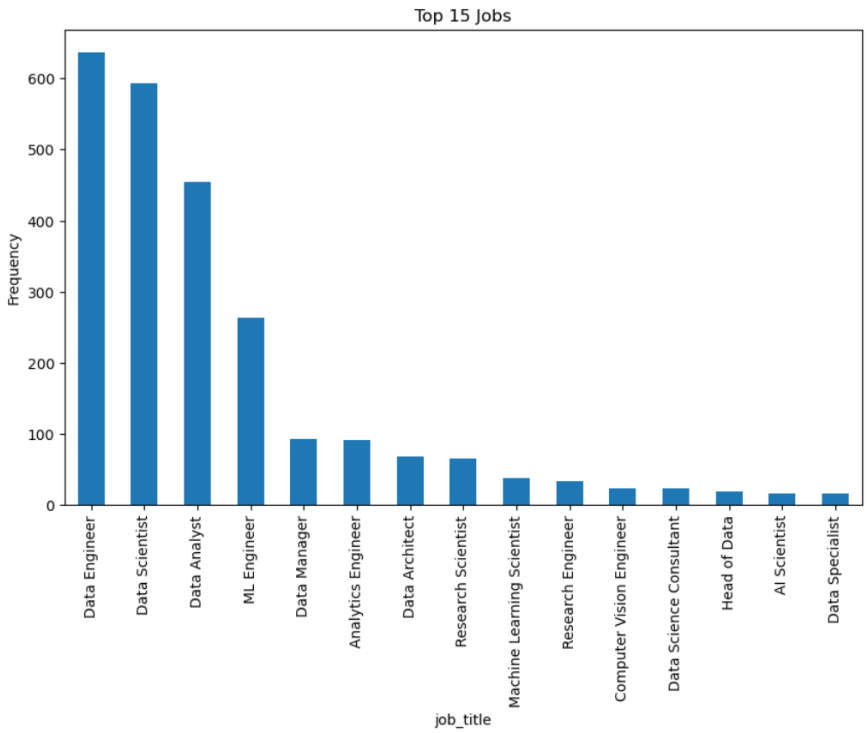


Figure 13: Plotting the bar graph for top 15 jobs

Before doing the plotting, we removed all the duplicate values from the data frame and stored it in a new data frame. Then we use value count to count the number of each job. After that we sort the values in descending order and use .head(15) to find out out the top 15 jobs. Then we plot it in the bar graph.

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

First, we sort the values of salary in USD in descending order, then we find out the top 5 highest salary jobs with their salary. After that we plot it in the bar graph.

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

```
In [35]: sort_salaries=cdf.sort_values(by='salary_in_usd', ascending=False)
         top_5_jobs = sort_salaries['job_title'].head(5).tolist()
         top_5_salaries = sort_salaries['salary_in_usd'].head(5).tolist()

In [36]: plt.figure(figsize=(10, 6))
         plt.bar(top_5_jobs, top_5_salaries)
         plt.title('Top 5 Highest Salaries')
         plt.ylabel('Salary')
         plt.xlabel('Job_Title')
         plt.xticks(rotation=90)

Out[36]: ([0, 1, 2, 3, 4],
          [Text(0, 0, 'Research Scientist'),
           Text(1, 0, 'Data Analyst'),
           Text(2, 0, 'AI Scientist'),
           Text(3, 0, 'Machine Learning Scientist'),
           Text(4, 0, 'Data Scientist')])
```

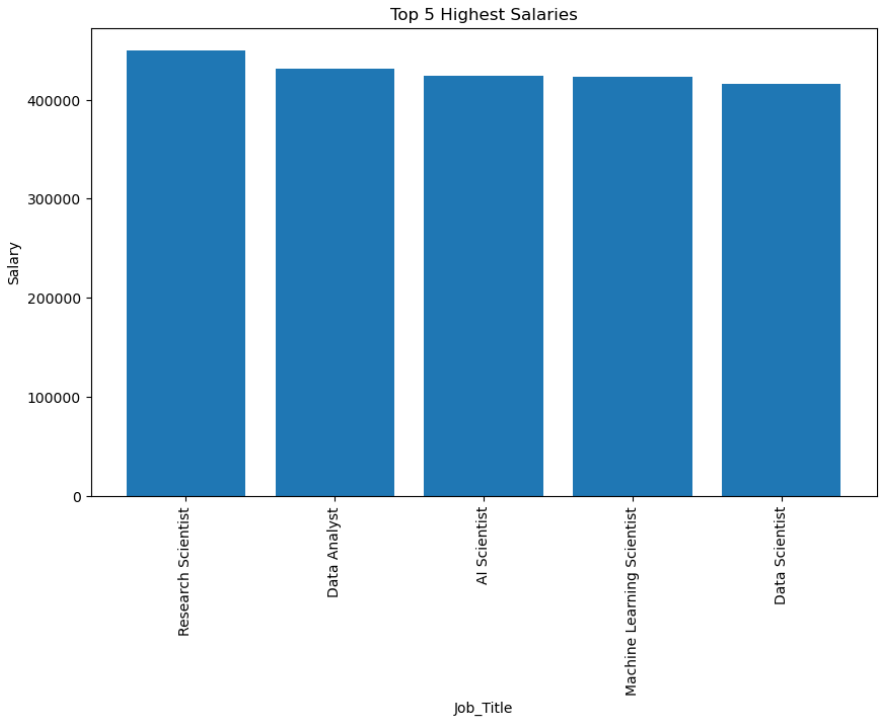


Figure 14: Plotting the top 5 paying jobs in bar graph

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

First, we group the data by experience level. Then, for each group, we find the employee with the highest salary ("max salary"). Finally, we separate the experience levels and the highest salaries into separate lists for easier use.

After which we plot the resulting data in a bar graph.

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

```
In [37]: experience_salary = cdf.groupby('experience_level')['salary_in_usd'].max()
exp_levels = experience_salary.index.to_numpy()
salaries = experience_salary.to_numpy()
plt.figure(figsize=(8, 6))
plt.bar(exp_levels, experience_salary)
plt.xlabel("Experience Level")
plt.ylabel("Salary (USD)")
plt.title("Salaries Based on Experience Level")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

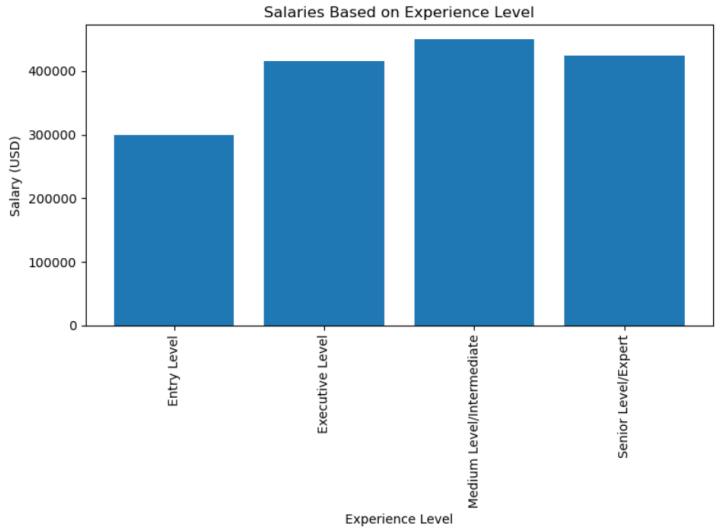


Figure 15: Plotting the salaries based on experience level

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

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

```
In [70]: plt.hist(cdf['salary_in_usd'], label='Salary (USD)')
plt.title("Histogram of Salary (USD)")
plt.xlabel("Salary")
plt.xticks(rotation=90)
plt.ylabel("Frequency")
plt.legend()
plt.show()
```

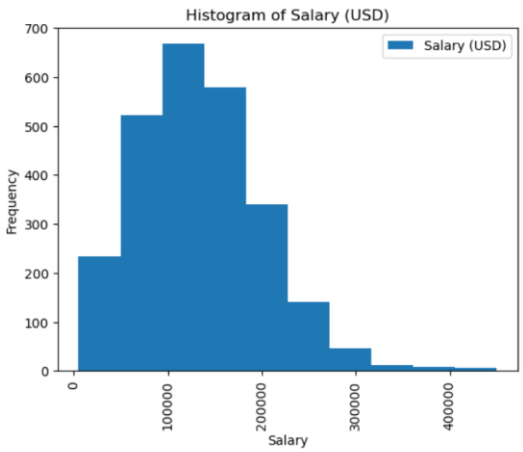


Figure 16: Histogram of work year and salary in USD

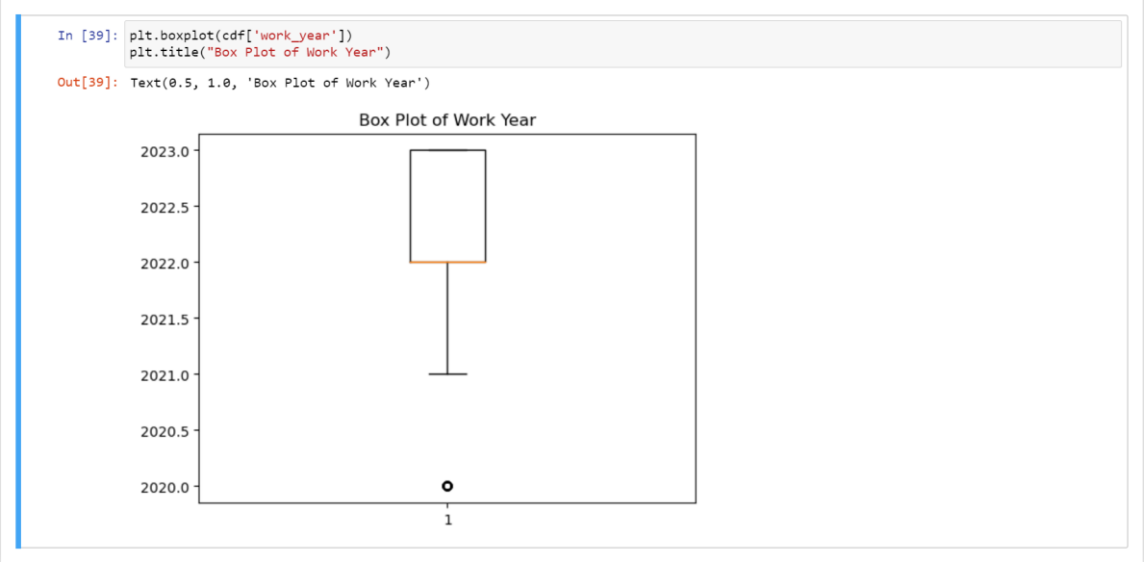


Figure 17: Box plot of work year

First, we plot the histogram of salary in USD then we box plot work year.