



CC5061NI

60% Individual Coursework

2023-24 Autumn

Student Name: Aakarshan Khadka

London Met ID: 22085855

College ID: NP01AI4S230002

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Monday, May 13, 2024

Word Count: [Click or tap here to enter text.](#)

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

1. Libraries Used.....	4
1.1. Pandas.....	4
1.2. NumPy	5
1.3. Matplotlib.....	5
2. Data Understanding.....	6
3. Data Preparation.....	9
3.1. Question number one.....	9
3.2. Question number two.	10
3.3. Question number three.	11
3.4. Question number four.....	12
3.5. Question number five.	14
3.6. Question number six.	16
4. Data Analysis	17
4.1. Question number One.....	17
4.2. Question number Two.	18
5. Data Exploration	19
5.1. Question number One.....	19
5.2. Question number Two.	21
5.3. Question number Three.	22
5.4. Question number Four.	24

Table of Figures

Figure 1: Importing Necessary Libraries.....	5
Figure 2: Data Example.....	8
Figure 3:Loading Values in the Data Frame	9
Figure 4: Using head() method.....	9
Figure 5: Dropping specified columns.	10
Figure 6: Checking Dropped Columns.	10
Figure 7: Dropping NaN Values.....	11
Figure 8: Checking for Duplicates.	12
Figure 9: Number of Duplicate Rows.....	13
Figure 10: Unique Values in Column 1.	14
Figure 11: Unique Values in Column 2.	15
Figure 12: Number of Unique Values in Column.	15
Figure 13: Renaming values as given.	16
Figure 14: Renaming done successfully.....	16
Figure 15: Showing Summary Statistics.....	17
Figure 16: Correlation between Variables	18
Figure 17: Computing the graph 5.1.....	19
Figure 18:Top 15 jobs.....	20
Figure 19: Computing Graph 5.2.....	21
Figure 20: Computing the graph 5.3.....	22
Figure 21: Salary Based on Experience Level	23

Figure 22: Histogram.....	24
Figure 23: Histogram of salary_in_usd.....	24
Figure 24:Box Plot.....	25
Figure 25: Box Plot of salary_in_usd.....	25

1. Libraries Used

For the completion of this coursework, I took help of 3 libraries:

1.1. Pandas

Pandas is an open-source python package that is most widely used for data science/data analysis and for machine learning tasks. It is built on top of another python package NumPy. Pandas is one of the most popular data wrangling packages. (S, 2020)

With pandas you can do a magnitude of different tasks which might have been time consuming and repetitive associated with data. These tasks may include:

- Data cleansing
- Data merging
- Data visualization

- Data loading and saving etc.

1.2. NumPy

NumPy is an open-source python library that provides a multidimensional array object. It also provides a vast assortment of routines and functions for fast operations on given array which includes mathematical, logical, shape manipulation, selection, sorting and I/O functions. You can also perform random stimulations, basic algebra, and statistical operations and much more. (NumPy, n.d.)

1.3. Matplotlib

Matplotlib is a powerful Python library for creating static, animated, and interactive visualizations. According to themselves, matplotlib makes easy thing easier and hard things possible. Matplotlib was first made by John Hunter; a neurobiologist to work with EEG data. (Matplotlib, n.d.).

Before continuing with the operations below, all the necessary libraries were imported as the first line in the kernel.

```
#importing the necessary librarires into the code.  
import pandas as pd  
import matplotlib.pyplot as plt  
import numpy as np
```

Figure 1: Importing Necessary Libraries

2. Data Understanding

S.No.	Column Name	Description	Data Type	Dataframe Datatype
1	work_year	How many years of work experience.	Integer	Integar
2	experience_level	Level of experience for the work.(e.g. Entry, Medium, Senior)	String	object
3	employment_type	What type of employment are they on. (e.g. Full-time, Part-time)	String	object
4	job_title	Title of the job of the employee	String	object
5	salary	Salary of employee	Numeric	integar

6	salary_currency	Currency of the salary of the employee	String	object
7	salary_in_usd	The employee salary in USD	Numeric	object
8	employee_residence	The Address/Residence of the employee	String	object
9	remote_ratio	Ratio of remote work.	Numeric	integar
10	company_location	Location of the company.	String	object
11	company_size	Size of the company	String	object

The dataset provided contains salary information related to the field of data science. It includes various attributes such as work year, experience level, job title, employment type, salary, employee residence, company location, and size. To effectively utilize this dataset for analysis, we need to start with data preparation. This involves tasks like handling missing values, ensuring data consistency, and accuracy. Once the data is prepared, we move on to data analysis, where we can apply statistical techniques and algorithms to find the correlations and develop insights into the factors that may influence the salaries. Finally, data exploration involves visualizing the dataset using graphs, charts, and statistical summaries which helps us to gain a comprehensive understanding of its structure. This process of understanding, preparing, analyzing, and exploring the data is crucial for deriving meaningful insights and achieving the objectives of the coursework.

A	B	C	D	E	F	G	H	I	J	K
2021	EN	FT	Data Science Consultant	90000	USD	90000	US	100	US	S
2021	MI	FT	Data Scientist	52000	EUR	61467	DE	50	AT	M
2021	SE	FT	Machine Learning Infrastructure Engineer	195000	USD	195000	US	100	US	M
2021	MI	FT	Data Scientist	32000	EUR	37825	ES	100	ES	L
2020	MI	FT	Data Analyst	85000	USD	85000	US	100	US	L
2021	EX	CT	Principal Data Scientist	416000	USD	416000	US	100	US	S
2021	SE	FT	Machine Learning Scientist	225000	USD	225000	US	100	CA	L
2021	MI	FT	Data Scientist	40900	GBP	56256	GB	50	GB	L
2021	MI	FT	Data Scientist	2500000	INR	33808	IN	0	IN	M
2021	MI	FT	Data Scientist	85000	GBP	116914	GB	50	GB	L
2021	MI	FT	Machine Learning Engineer	180000	PLN	46597	PL	100	PL	L
2020	MI	FT	Data Analyst	8000	USD	8000	PK	50	PK	L
2020	EN	FT	Data Engineer	4450000	JPY	41689	JP	100	JP	S
2020	SE	FT	Big Data Engineer	100000	EUR	114047	PL	100	GB	S
2021	MI	FT	Machine Learning Engineer	75000	EUR	88654	BE	100	BE	M
2020	EN	FT	Data Science Consultant	423000	INR	5707	IN	50	IN	M
2020	MI	FT	Lead Data Engineer	56000	USD	56000	PT	100	US	M
2021	EN	PT	Computer Vision Engineer	180000	DKK	28609	DK	50	DK	S
2021	MI	FT	Data Scientist	75000	EUR	88654	DE	50	DE	L
2020	MI	FT	Product Data Analyst	450000	INR	6072	IN	100	IN	L
2020	SE	FT	Data Engineer	42000	EUR	47899	GR	50	GR	L
2020	MI	FT	BI Data Analyst	98000	USD	98000	US	0	US	M
2021	MI	FT	Data Engineer	48000	GBP	66022	HK	50	GB	S
2021	MI	FT	Research Scientist	48000	EUR	56738	FR	50	FR	S

Figure 2: Data Example

3. Data Preparation

3.1. Question number one

Write a python program to load data into pandas DataFrame.

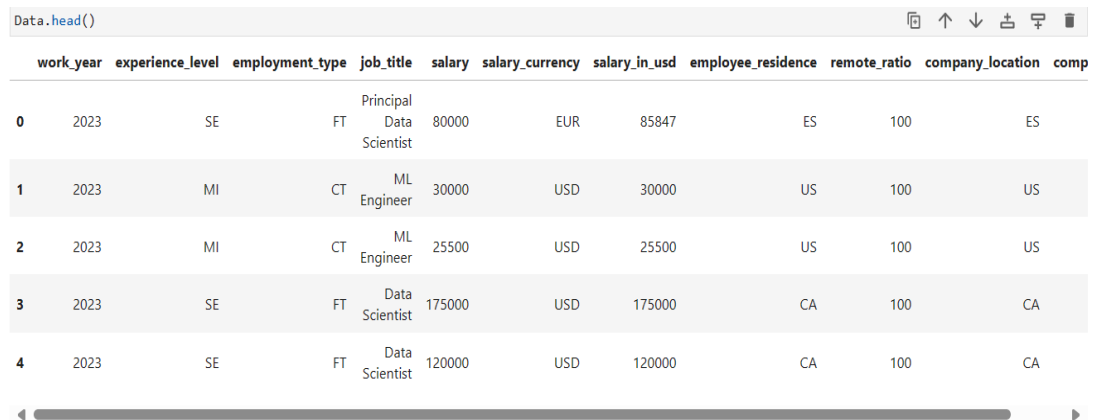
To load the given data into the data frame, pandas library was used.

```
Data = pd.read_csv('salary.csv')
```

Figure 3:Loading Values in the Data Frame

This line makes use of the built-in method of the pandas library. It is specifically designed to read data from CSV (Comma Separated Values) files into a DataFrame, which is one of the primary data structures in Pandas.

After loading the DataFrame into the kernel, the `.head()` method is used to check if the data was successfully loaded; This method returns the first few rows of the DataFrame, by default it is set to 5.



	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	comp
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	

Figure 4: Using head() method.

3.2. Question number two.

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

To remove the unnecessary columns; in this case salary and salary currency the `.drop()` method is used. This method takes three parameters. `["salary", "salary_currency"]` which specifies the columns to be dropped. `"axis = 1"` specifies that the operations will be performed upon the column axis and `inplace=True` which says that this DataFrame is to be modified upon i.e. no creation of a new DataFrame.

```
Data.drop(["salary", "salary_currency"],axis = 1 , inplace = True)
```

Figure 5: Dropping specified columns.

After dropping, we are to check if the operation was completed.

```
Data.columns
```

```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',  
      'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',  
      'remote_ratio', 'company_location', 'company_size'],  
      dtype='object')
```

```
Data.drop(["salary", "salary_currency"],axis = 1 , inplace = True)
```

```
Data.columns
```

```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',  
      'salary_in_usd', 'employee_residence', 'remote_ratio',  
      'company_location', 'company_size'],  
      dtype='object')
```

Figure 6: Checking Dropped Columns.

3.3. Question number three.

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

To remove the NaN values from the uploaded DataFrame (Data), the `.dropna()` method was used with the parameter `inplace = True`. This method is part of the pandas library which was designed to remove rows containing missing values in a DataFrame. The parameter `inplace= True` is used to ensure that the changes are made and saved into the original DataFrame instead of making a new one.

```
#Drop rows with NaN missing values.  
Data.dropna(inplace=True)
```

```
Data.isna()
```

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...
3750	False	False	False	False	False	False	False	False	False
3751	False	False	False	False	False	False	False	False	False
3752	False	False	False	False	False	False	False	False	False
3753	False	False	False	False	False	False	False	False	False
3754	False	False	False	False	False	False	False	False	False

3755 rows × 9 columns

Figure 7: Dropping NaN Values.

After dropping the NaN value the `.isna()` method is used to check if the DataFrame has NaN values or not. As all the rows return false; the operation was completed successfully.

3.4. Question number four.

Write a python program to check duplicates value in the DataFrame.

To check for duplicates values, the `.duplicated()` method is used. This method creates a Boolean series named `duplicate`; The method indicates if the row is a duplicate or not.

```
#check for duplicates row:  
duplicate = Data.duplicated()  
duplicate
```

```
0      False  
1      False  
2      False  
3      False  
4      False  
...  
3750    False  
3751    False  
3752    False  
3753    False  
3754    False  
Length: 3755, dtype: bool
```

Figure 8: Checking for Duplicates.

Afterwards, the `sum()` method is used which will calculate the sum of variables in a DataFrame. Since the `.duplicated` method returns a Boolean value i.e. 0 for False and 1 for True; `sum()` allows us to see how many duplicate rows are present.

```
num_duplicates = Data.duplicated().sum()  
print("Number of duplicates:", num_duplicates)
```

Number of duplicates: 1171

Figure 9: Number of Duplicate Rows.

3.5. Question number five.

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

To see the number of unique values from each column in the DataFrame, the `.unique()` method was used. It returns an array of unique value from the column. A for loop is used to iterate over the columns.

```
#Iterating over each column too see the unique values.
for column in Data.columns:
    print(column, ":", Data[column].unique())

work_year : [2023 2022 2020 2021]
experience_level : ['SE' 'MI' 'EN' 'EX']
employment_type : ['FT' 'CT' 'FL' 'PT']
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'
'Data Science Manager' 'Data Manager' 'Machine Learning Researcher'
'Big Data Engineer' 'Data Specialist' 'Lead Data Analyst'
'BI Data Engineer' 'Director of Data Science'
'Machine Learning Scientist' 'MLOps Engineer' 'AI Scientist'
'Autonomous Vehicle Technician' 'Applied Machine Learning Scientist'
'Lead Data Scientist' 'Cloud Database Engineer' 'Financial Data Analyst'
'Data Infrastructure Engineer' 'Software Data Engineer' 'AI Programmer'
'Data Operations Engineer' 'BI Developer' 'Data Science Lead'
'Deep Learning Researcher' 'BI Analyst' 'Data Science Consultant'
'Data Analytics Specialist' 'Machine Learning Infrastructure Engineer'
'BI Data Analyst' 'Head of Data Science' 'Insight Analyst'
'Deep Learning Engineer' 'Machine Learning Software Engineer'
'Big Data Architect' 'Product Data Analyst'
'Computer Vision Software Engineer' 'Azure Data Engineer'
'Marketing Data Engineer' 'Data Analytics Lead' 'Data Lead'
'Data Science Engineer' 'Machine Learning Research Engineer'
'NLP Engineer' 'Manager Data Management' 'Machine Learning Developer'
'3D Computer Vision Researcher' 'Principal Machine Learning Engineer'
'Data Analytics Engineer' 'Data Analytics Consultant'
'Data Management Specialist' 'Data Science Tech Lead'
'Data Scientist Lead' 'Cloud Data Engineer' 'Data Operations Analyst'
'Marketing Data Analyst' 'Power BI Developer' 'Product Data Scientist'
'Principal Data Architect' 'Machine Learning Manager'
'Lead Machine Learning Engineer' 'ETL Developer' 'Cloud Data Architect'
'Lead Data Engineer' 'Head of Machine Learning' 'Principal Data Analyst'
'Principal Data Engineer' 'Staff Data Scientist' 'Finance Data Analyst']
```

Figure 10: Unique Values in Column 1.

```

salary_in_usd : [ 85847  30000  25500 ...  28369 412000  94665]
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']
remote_ratio : [100  0  50]
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']
company_size : ['L' 'S' 'M']

```

Figure 11: Unique Values in Column 2.

Additionally, the `nunique()` method returns the number of unique values from each column.

```

#To count the number of unique value in each column.
Data.nunique()

```

```

work_year          4
experience_level    4
employment_type     4
job_title          93
salary_in_usd      1035
employee_residence  78
remote_ratio        3
company_location    72
company_size        3
dtype: int64

```

Figure 12: Number of Unique Values in Column.

3.6. Question number six.

Rename the experience level columns as below:

SE – Senior Level/Expert

MI – Medium Level/Intermediate

EN – Entry Level

EX – Executive Level

To rename the experience level in the columns as given, the where function of Numpy library was used. It performs element-wise conditional operations.

The syntax goes : `np.where(condition, x, y)` where x is the value assigned if the condition is met and y the value when the condition is not met.

```
Data["experience_level"] = np.where(Data["experience_level"] == "SE", "Senior Level/Expert",
    np.where(Data["experience_level"] == "MI", "Medium Level/Intermediate",
    np.where(Data["experience_level"] == "EN", "Entry Level",
    np.where(Data["experience_level"] == "EX", "Executive Level", Data["experience_level"]))))
```

Figure 13: Renaming values as given.

Afterwards, the `head()` method is used to see if the operation took place.

Data.head()									
	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	Principal 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

Figure 14: Renaming done successfully.

4. Data Analysis

4.1. Question number One.

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

The variable chosen was "salary_in_usd". To calculate the sum, mean, standard deviation, skewness and kurtosis of the variable the following were used:

- **sum():** Calculates the sum of the values in a Series or DataFrame.
- **std():** Calculates the standard deviation of the values in a Series or DataFrame.
- **mean():** Calculates the mean (average) of the values in a Series or DataFrame.
- **skew():** Calculates the skewness of the values in a Series or DataFrame.
- **kurt():** Calculates the kurtosis of the values in a Series or DataFrame.

```
variable = 'salary_in_usd'
sumValue=Data[variable].sum()
stdValue=Data[variable].std()
meanValue=Data[variable].mean()
skewness = Data[variable].skew()
kurtosis = Data[variable].kurt()

print("Sum of", variable, ":", sumValue)
print("Standard deviation of", variable, ":", stdValue)
print("Mean of", variable, ":", meanValue)
print("Skewness of", variable, ":", skewness)
print("Kurtosis of", variable, ":", kurtosis)

Sum of salary_in_usd : 516576814
Standard deviation of salary_in_usd : 63055.625278224084
Mean of salary_in_usd : 137570.38988015978
Skewness of salary_in_usd : 0.5364011659712974
Kurtosis of salary_in_usd : 0.8340064594833612
```

Figure 15: Showing Summary Statistics.

4.2. Question number Two.

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

To find the correlation of all variables, the `.corr()` method was used. The `'corr()'` method calculates the correlation matrix of all numeric variables in the DataFrame. As there are some values that aren't integer i.e. they can't be calculated for in correlation, the parameter `"numeric_only = True"` was used that ensure that the values taken are all numeric.

```
Data.corr(numeric_only=True)
```

	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 16: Correlation between Variables

5. Data Exploration

5.1. Question number One.

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

To find the top 15 jobs, the `value_counts()` method was used. The `head(15)` ensures that the only the top 15 jobs is taken.

```
topJobs = Data['job_title'].value_counts().head(15)

plt.figure(figsize=(10, 6))
topJobs.plot(kind='bar')
plt.title('Top 15 Jobs')
plt.xlabel('Job Title')
plt.ylabel('Frequency')
plt.xticks(rotation=90)
plt.show()
```

Figure 17: Computing the graph 5.1.

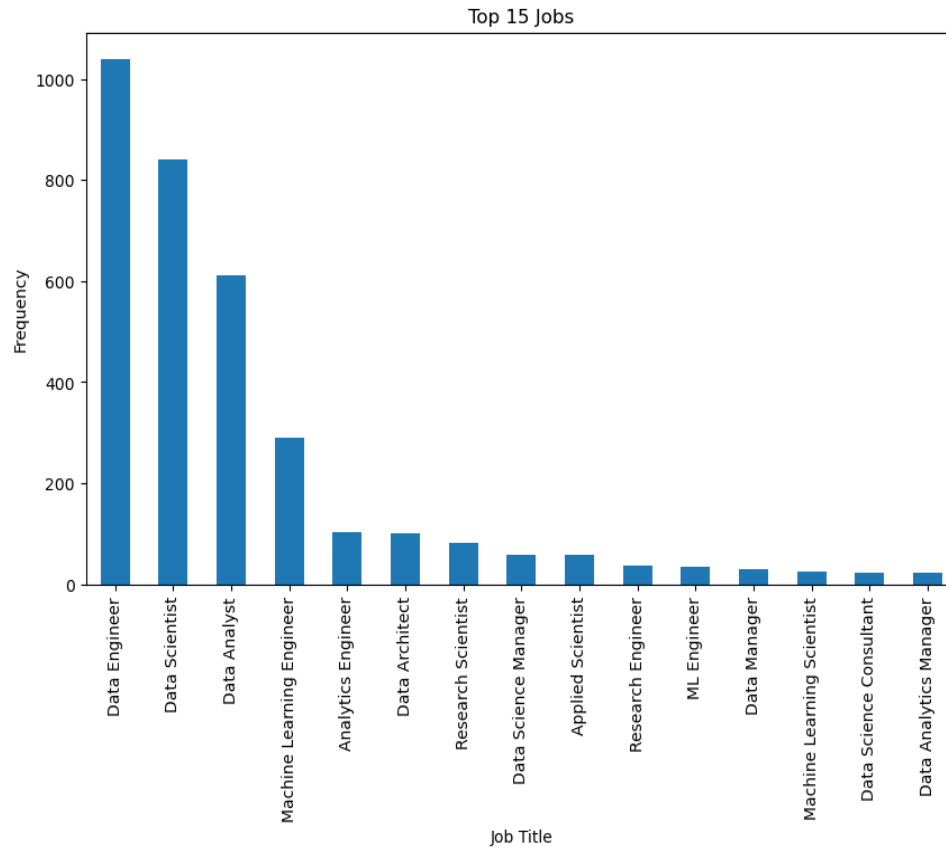


Figure 18: Top 15 jobs.

- **plt.figure:** This method creates a new figure for plotting. The fig size parameter specifies the size of the figure, with (10, 6) indicating 10 inches in width and 6 inches in height.
- **plt.title:** This method sets the title of the plot to "Top 15 Jobs".
- **plt.xlabel:** This method sets the label for the x-axis to "Job Title".
- **plt.ylabel:** This method sets the label for the y-axis to "Frequency".
- **plt.xticks:** This method rotates the x-axis tick labels by 90 degrees for better readability.
- **plt.show:** This method displays the plot.

5.2. Question number Two.

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

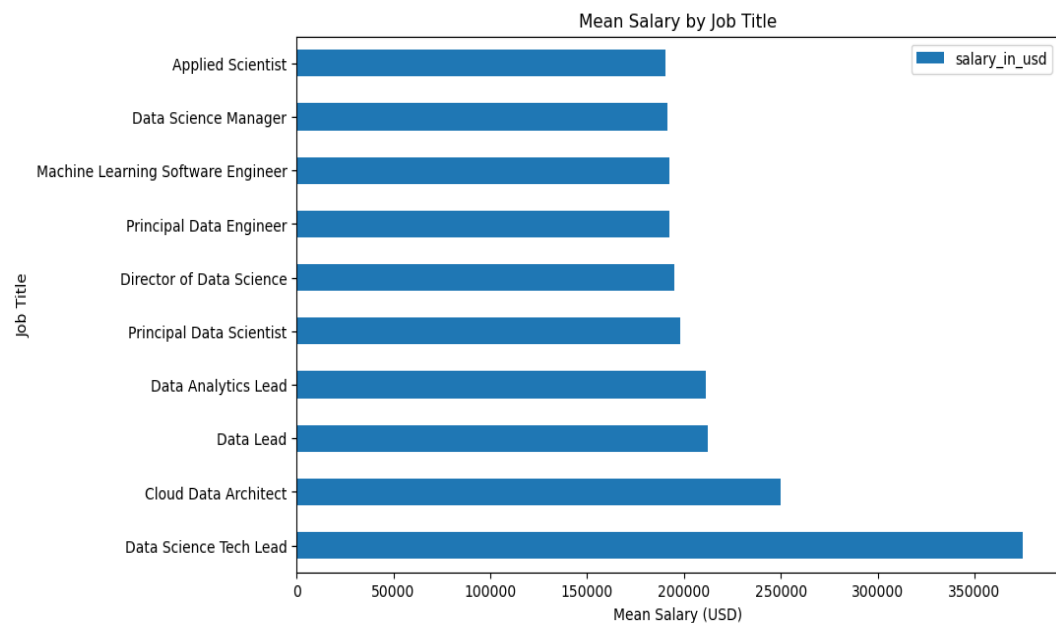
```
# Group data by job title and calculate mean salary for each job
mean_salary_by_job = Data.groupby('job_title')['salary_in_usd'].mean().sort_values(ascending=False).head(10)

# Identify job with the highest mean salary
highest_salary_job = mean_salary_by_job.idxmax()
highest_salary = mean_salary_by_job.max()

# Plot bar graph
plt.figure(figsize=(10, 6))
mean_salary_by_job.plot(kind='barh')
plt.title('Mean Salary by Job Title')
plt.xlabel('Mean Salary (USD)')
plt.ylabel('Job Title')
plt.xticks(rotation=0)
plt.legend()
plt.show()
```

Figure 19: Computing Graph 5.2.

To find the job with the highest salaries, the data was first grouped by their job titles. Afterwards, the mean of their respective salary was taken, the calculated mean was then placed in descending order and the head() method was used to take the top 10 highest salaries. The graph was then plotted.



- **plt.figure(figsize=(10, 6))**: This line creates a new figure for plotting with a specified size of 10 inches by 6 inches.
- **mean_salary_by_job.plot(kind='barh')**: This line creates a horizontal bar plot of the mean salary for each job title.
- **plt.title('Mean Salary by Job Title')**: This line sets the title of the plot as "Mean Salary by Job Title".
- **plt.xlabel('Mean Salary (USD)')**: This line sets the label for the x-axis as "Mean Salary (USD)".
- **plt.ylabel('Job Title')**: This line sets the label for the y-axis as "Job Title".
- **plt.xticks(rotation=0)**: This line sets the rotation of the x-axis tick labels to 0 degrees.
- **plt.show()**: This line displays the plot.

5.3. Question number Three.

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

To calculate the salaries based on experience level, the data is first grouped by experience level using the groupby() method and then the mean() is used to calculate the mean of the salaries for an accurate graph.

```
mean_salary_by_experience_level = Data.groupby('experience_level')['salary_in_usd'].mean()

plt.figure(figsize=(10, 6))
mean_salary_by_experience_level.plot(kind='bar')
plt.title('Salary based Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Mean Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```

Figure 20: Computing the graph 5.3.



Figure 21: Salary Based on Experience Level

- **plt.title:** This method sets the title of the plot to "Salary Based Experience Level".
- **plt.xlabel:** This method sets the label for the x-axis to "Experience Level".
- **plt.ylabel:** This method sets the label for the y-axis to "Mean Salary (USD)".
- **plt.xticks:** This method rotates the x-axis tick labels for better readability.
- **plt.show:** This method displays the plot.

5.4. Question number Four.

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

The chosen variables for both the graphs were “salary_in_usd”.

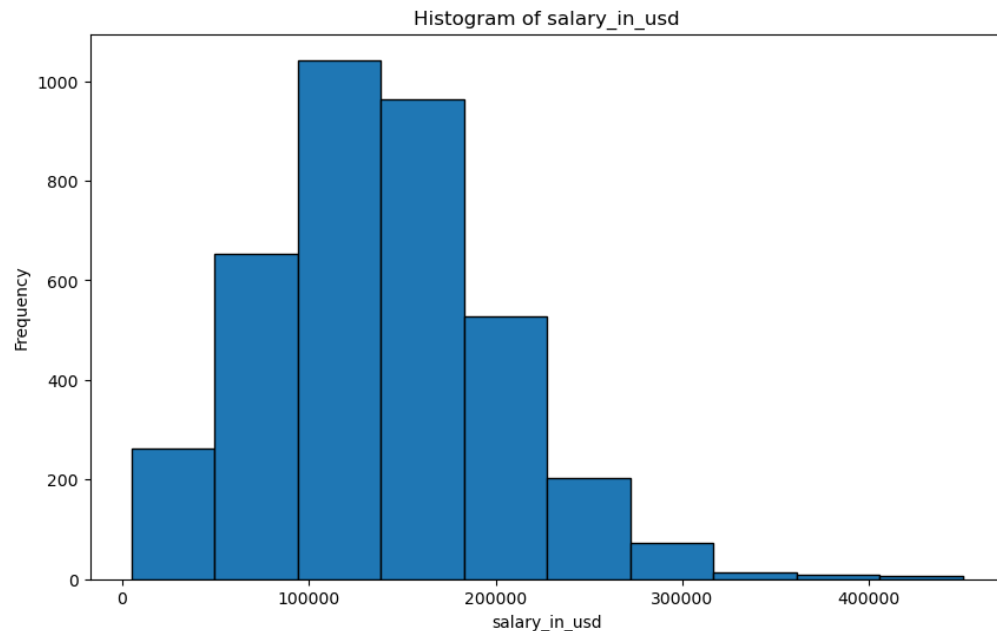


Figure 22: Histogram

Figure 23: Histogram of salary_in_usd

- **plt.figure(figsize=(10, 6))**: This line creates a new figure for plotting with a specified size of 10 inches by 6 inches.
- **plt.hist()**: This line creates a histogram plot of the chosen variable from the DataFrame. The bins parameter specifies the number of bins, and edgecolor='black' sets the color of the edges of the bars to black for better visibility.
- **plt.title()**: This line sets the title of the histogram plot dynamically using the chosen_variable.
- **plt.xlabel()**: This line sets the label for the x-axis of the histogram plot as the chosen variable.

- **plt.ylabel():** This line sets the label for the y-axis of the histogram plot as "Frequency".
- **plt.show():** This line displays the histogram plot.

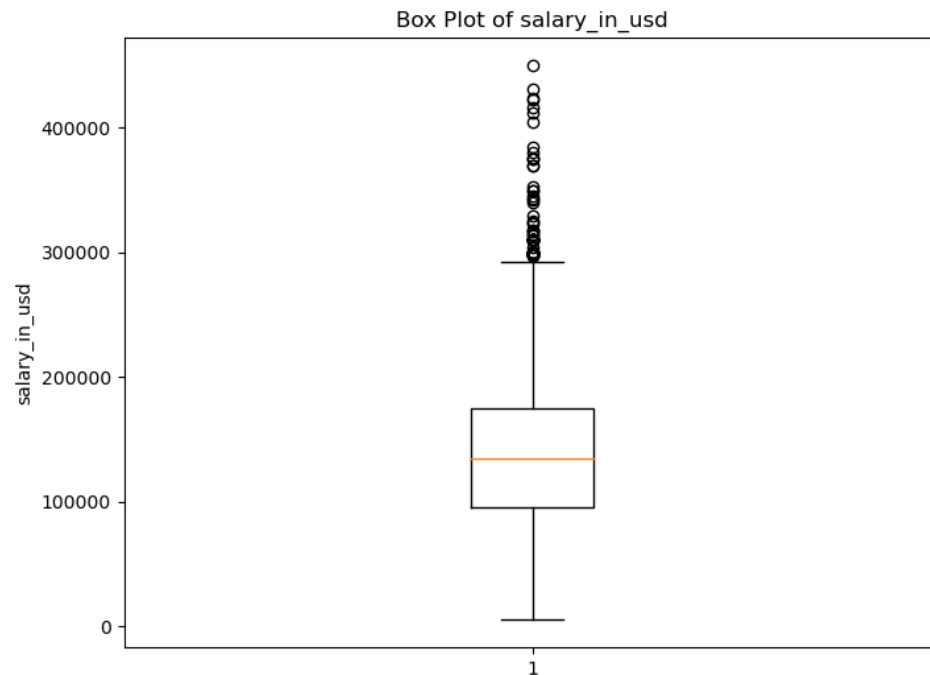


Figure 24: Box Plot

Figure 25: Box Plot of salary_in_usd.

- **plt.figure(figsize=(8, 6)):** This line creates a new figure for plotting with a specified size of 8 inches by 6 inches.
- **plt.boxplot():** This line creates a box plot of the chosen variable from the DataFrame.
- **plt.title():** This line sets the title of the box plot dynamically using the chosen variable.
- **plt.ylabel():** This line sets the label for the y-axis of the box plot as the chosen variable.
- **plt.show():** This line displays the box plot.

References

Matplotlib, n.d.. *Matplotlib*. [Online]
Available at: <https://matplotlib.org/>
[Accessed 12 5 2024].

NumPy, n.d.. *NumPy*. [Online]
Available at: <https://numpy.org/doc/stable/user/whatisnumpy.html>
[Accessed 12 5 2024].

S, S., 2020. *What Is Pandas in Python? Everything You Need to Know*. [Online]
Available at: <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>
[Accessed 12 5 2024].