



CC5067NI-Smart Data Discovery

60% Individual Coursework

2023-24 Spring

Student Name: Krish Bhattarai

London Met ID: 22067570

College ID: np01cp4a220071@islingtoncollege.edu.np

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Monday, May 13, 2024

Word Count: 1692

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.

Table of Contents

1. Introduction	1
2. Data Understanding.....	2
3. Data Preparation.....	4
a. Write a python program to load data into pandas DataFrame.	4
b. Write a python program to remove unnecessary columns i.e., salary and salary currency.	5
c. Write a python program to remove the NaN missing values from updated dataframe.	6
d. Write a python program to check duplicate values in the dataframe.....	7
e. Write a python program to see the unique values from all the columns in the dataframe.	9
f. Rename the experience level columns as below.	12
Data Cleaning	16
4. Data Analysis	17
a. Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.....	17
b. Write a Python program to calculate and show correlation of all variables.	18
5. Data Exploration	19
a. Write a python program to find out top 15 jobs. Make a bar graph of sales as well.	19
b. Which job has the highest salaries? Illustrate with bar graph.	21
c. Write a python program to find out salaries based on experience level. Illustrate it through bar graph.	23
d. Write a Python program to show histogram and box plot of any chosen different variables. Use proper labels in the graph.	25

6. Conclusion.....	29
--------------------	----

Table of Tables

Table 1: Data understanding and Column description.	3
--	---

Table of Figures

Figure 1: Load Data into pandas Dataframe.	4
Figure 2: Removing the unnecessary columns.	5
Figure 3: Checking if the columns have been removed.....	5
Figure 4: Removing the NaN missing values.	6
Figure 5: Checking for duplicated in dataframe.....	7
Figure 6: Removing Duplicates.	8
Figure 7: Displaying the unique values from the dataframe.	10
Figure 8: Displaying the unique values from the dataframe.	11
Figure 9: Renaming SE to Senior Level in the column experience_level.	12
Figure 10: Renaming ML to Medium Level in the column experience_level.....	13
Figure 11: Renaming EN to Entry Level in the column experience_level.	14
Figure 12: Renaming EX to Executive Level in the column experience_level.	15
Figure 13: Data Cleaning.	16
Figure 14: Display summary stats.	17
Figure 15: calculate and show correlation of variables.....	18
Figure 16: Find top 15 jobs.....	19
Figure 17: Display top 15 jobs in bargraph.....	20
Figure 18: Job with highest salary.	21
Figure 19: Display job with highest salary.	21
Figure 20: Display jobs with highest salary in a bargraph.	22
Figure 21: Salaries based on experience.	23
Figure 22: Display salaries based on experience in a bargraph.....	24
Figure 23: Plot histogram.	25
Figure 24: Display histogram.....	26
Figure 25: code for displaying boxplot Display boxplot.....	27
Figure 26: Display boxplot.....	28

1. Introduction

The coursework involves understanding and exploring a dataset which contains various information about employees. This includes data such as their salary, residence, company size, job title, etc. The coursework requires exploration of the data. For this the data must be inspected first. Understanding the data is very important as it uncovers many doubts and queries. The data is then cleaned as it contains null data, inconsistent data, which decreases efficiency. This provides better skills on how data is cleaned to be explored in a system. This helps with optimized data preparation and data mining. Feature Engineering is done, meaning that the data is changed for better understanding of the data. Graphical representation is created allowing for data to be visualized. Patterns, similarities, and relationships in the data is present which must be overcome as data redundancy may hamper with data exploration.

2. Data Understanding

The data provided is presented in a spreadsheet in tabular format. The file contains various data about the employees. The data is separated by columns which represents data such as work year, experience level, employment type, job title, salary, salary in currency, salary in usd, employee residence, remote ratio, company location and company size. Work year contains data in integer, it contains data about when the employee joined. Experience level is an object data type which contains data about the experience level of the employees, Employee type contains data about the type of employee. Job title is an object data type which contains data about what job an employee has. Salary column contains the annual salary of the employees. Salary currency column contains data about the type of currency the employees earn. Salary in usd column is an integer data type which contains data about the salary of the employees in US dollar. The employee residence is an object data type, it contains data about which country the employees reside. The company location contains data about which country the company is located in. The company size is an object data type which contains data about the size of the company in which the employees work.

The various data provides information about how the job title and experience level influences the salary of the employee. Data cleaning must be done as it contains data inconsistencies. Cleaning the data is important as it improves the accuracy, consistency, and quality of data. This influences the output of the data when exploring making the exploration phase more efficient.

s.No	Column Name	Description	Datatype
1	work_year	This column contains the data about when the employee joined the job.	int64
2	experience_level	This column contains data about the experience level of the employee.	object
3	employment_type	This column contains about the type of the employee.	object
4	job_title	This column contains data about what job the employee does. It displays the job title of the employee.	object
5	salary	This column contains the salary of the employee. It contains the salary information of some employee in different currency.	int64
6	salary_currency	This column contains data about what currency the employee earns in.	object
7	salary_in_usd	This column contains data about the annual salary of the employee in usd.	int64
8	employee_residence	This column contains data about what country the employee resides in.	object
9	remote_ratio	This column contains data about how likely the job is to be remote.	int64
10	company_location	This column contains data about which country the company is located in.	object
11	company_size	This column contains data about how small or large the company is.	object

Table 1: Data understanding and Column description.

3. Data Preparation

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

Answer:

Creating a dataframe 'sal', read a csv file named "DataScienceSalarie.csv". The file is then printed.

Write a python program to load data into pandas DataFrame

```
sal=pd.read_csv("DataScienceSalarie.csv") #creating a dataframe named sal
sal
```

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

3755 rows x 11 columns

Figure 1: Load Data into pandas Dataframe.

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

Answer:

'unnecessary' contains columns 'salary' and 'salary_currency' which aren't necessary in the dataframe. The columns are dropped and then printed.

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

```
unnecessary = ['salary', 'salary_currency'] #columns to remove
sal = sal.drop(columns = unnecessary) #dropping tables
sal
```

	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.

Displaying the columns to check if the unnecessary columns have been removed.

```
sal.columns

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

Figure 3: Checking if the columns have been removed.

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

Answer:

The 'dropna()' method is used to remove the missing value from the 'sal' dataframe.

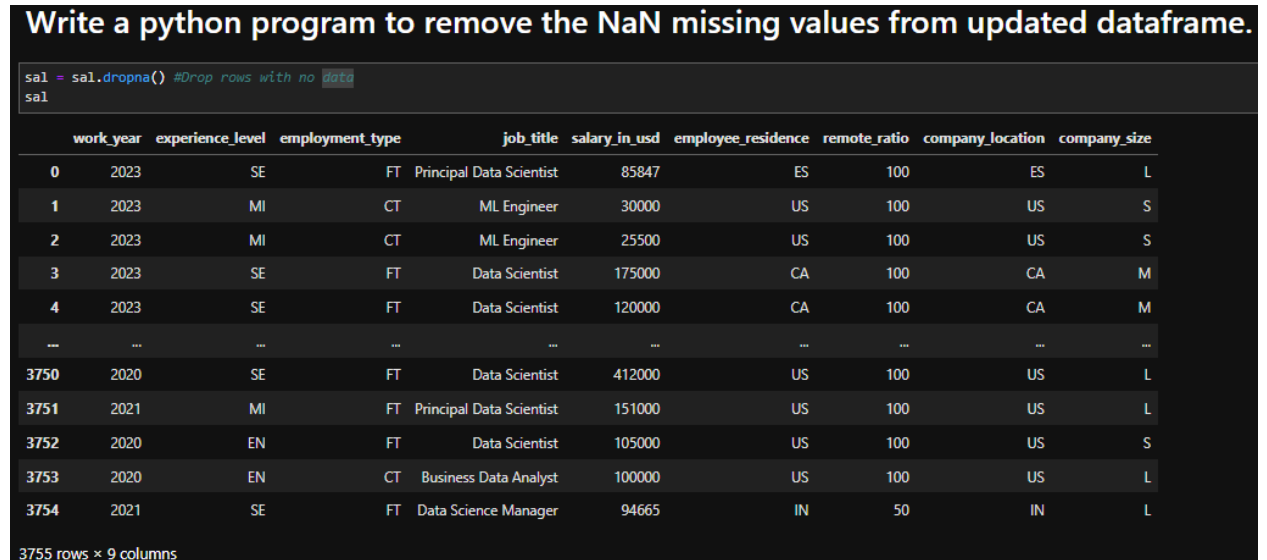


Figure 4: Removing the NaN missing values.

d. Write a python program to check duplicate values in the dataframe.

Answer:

The code uses '.duplicated()' method to check for duplicated values in the dataframe.

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

```
duplicateValues = sal[sal.duplicated()] #check duplicate rows
duplicateValues
```

	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

1171 rows × 9 columns

Figure 5: Checking for duplicated in dataframe.

The '.duplicated()' method is used to check for duplicate values. 'duplicated()==False' updates the dataframe to remove the duplicate values.

```
sal = sal[sal.duplicated()==False] #Removing duplicate data
sal
```

	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

2584 rows × 9 columns

Figure 6: Removing Duplicates.

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

Answer:

An empty dictionary 'unique_dict' is created. It stores unique values. The for loop in sal.columns iterates through the columns in the dataframe. A unique value is assigned to the key in 'unique_dict'. The '.unique()' method obtains unique values for each columns.

The for loop in unique.dict.items() is used to iterate through the key value pairs in 'unique_dict' dictionary.

The values are then printed.

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

```
unique_dict = {} #Creating an empty dictionary

for column in sal.columns: #Iterating through the dataframe
    unique_dict[column] = sal[column].unique()

for column, values in unique_dict.items(): #Displaying the unique values
    print(f'Unique Values {column}:') #print(f) to print the column name
    print(values)
    print()

Unique Values 'work_year':
[2023 2022 2020 2021]

Unique Values 'experience_level':
['SE' 'MI' 'EN' 'EX']

Unique Values 'employment_type':
['FT' 'CT' 'FL' 'PT']

Unique Values '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'
'MLP 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']
```

Figure 7: Displaying the unique values from the dataframe.

```

Principal Data Engineer | Staff Data Scientist | Finance Data Analyst |

Unique Values 'salary_in_usd':
[ 85847 30000 25500 ... 28369 412000 94665]

Unique Values '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 'remote_ratio':
[100  0 50]

Unique Values '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 'company_size':
['L' 'S' 'M']

```

Figure 8: Displaying the unique values from the dataframe.

f. Rename the experience level columns as below.

SE – Senior Level/Expert

Answer:

The '.replace' method replaces 'SE' to 'Senior Level' in the 'experience_level' column. The '.loc' method is used for label based indexing which access the rows or columns based on the label names.

Rename the experience level of the given columns

```
# SE - Senior Level/Expert
sal.loc[:, 'experience_level'] = sal['experience_level'].replace("SE", "Senior Level") #change SE to Senior Level
sal
```

	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
3754	2021	Senior Level	FT	Data Science Manager	94665	IN	50	IN	L

2584 rows x 9 columns

Figure 9: Renaming SE to Senior Level in the column experience_level.

MI – Medium Level/Intermediate

Answer:

The '.replace' method replaces 'MI' to 'Medium Level' in the 'experience_level' column. The '.loc' method is used for label based indexing which access the rows or columns based on the label names.

```
#MI - Medium Level/Intermediate
sal.loc[:, 'experience_level'] = sal['experience_level'].replace("MI", "Medium Level") #change MI to Medium Level
sal
```

	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
3754	2021	Senior Level	FT	Data Science Manager	94665	IN	50	IN	L

2584 rows × 9 columns

Figure 10: Renaming ML to Medium Level in the column experience_level.

EN – Entry Level

Answer:

The '.replace' method replaces 'EN' to 'Entry Level' in the 'experience_level' column. The '.loc' method is used for label based indexing which access the rows or columns based on the label names.

```
# EN - Entry Level
sal.loc[:, 'experience_level'] = sal['experience_level'].replace("EN", "Entry Level") #change EN to Entry Level
sal
```

	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

2584 rows x 9 columns

Figure 11: Renaming EN to Entry Level in the column experience_level.

EX – Executive Level

Answer:

The '.replace' method replaces 'EX' to 'Executive Level' in the 'experience_level' column. The '.loc' method is used for label based indexing which access the rows or columns based on the label names.

```
#EN – Executive Level
sal.loc[:, 'experience_level'] = sal['experience_level'].replace("EX", "Executive Level") #change SE to Executive Level
sal
```

	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

2584 rows × 9 columns

Figure 12: Renaming EX to Executive Level in the column experience_level.

Data Cleaning

A dictionary 'similar' is defined which maps data to its respective jobs. A new dictionary 'replaced' is created. A nested loop is performed which iterates over the specific job_titles. Each job title is then replaced with the specific jobs.

```

Data Cleaning

#Cleaning the data
# Define dictionary and map data to respective jobs
similar = {
    "Data Analytics Specialist": ["Financial Data Analyst", "Marketing Data Analyst", "Data Analytics Specialist"],
    "Data Scientist": ["Applied Data Scientist", "Staff Data Scientist", "Principal Data Scientist"],
    "Lead Data Scientist": ["Lead Data Scientist", "Lead Data Analyst", "Lead Machine Learning Engineer", "Lead Data Engineer"]
}

replaced = {replaced_title: similar_title
            for similar_title, replaced_titles in similar.items() #iterate through each key value pair in the dictionary similar_table
            for replaced_title in replaced_titles} #iterate through List of replaced titles for similar job titles
sal.loc[:, 'job_title'] = sal['job_title'].replace(replaced) # Replace job titles in the dataframe 'sal'

print(sal)

```

	work_year	experience_level	employment_type	job_title
0	2023	Senior Level	FT	Data Scientist
1	2023	Medium Level	CT	ML Engineer
2	2023	Medium Level	CT	ML Engineer
3	2023	Senior Level	FT	Data Scientist
4	2023	Senior Level	FT	Data Scientist
...
3750	2020	Senior Level	FT	Data Scientist
3751	2021	Medium Level	FT	Data Scientist
3752	2020	Entry Level	FT	Data Scientist
3753	2020	Entry Level	CT	Business Data Analyst
3754	2021	Senior Level	FT	Data Science Manager

	salary_in_usd	employee_residence	remote_ratio	company_location
0	85847	ES	100	ES
1	30000	US	100	US
2	25500	US	100	US
3	175000	CA	100	CA
4	120000	CA	100	CA
...
3750	412000	US	100	US
3751	151000	US	100	US
3752	105000	US	100	US
3753	100000	US	100	US
3754	94665	IN	50	IN

Figure 13: Data Cleaning.

4. Data Analysis

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

Answer:

A variable name is asked to be input. The value is then checked if the value is present in the dataframe. The entered value is then checked if it is numeric or not. If the correct variable is input, then checked if it is int 64 or float. If the entered variable is validated and is correct, the summary, skewness, kurtosis is displayed. If the prompt is incorrect, then a suitable error message is displayed.

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

variable = input("Enter the Variable name: ") #Enter variable name

if variable in sal.columns: #check for the variable name in the dataframe
    if sal[variable].dtype in ['int64', 'float64']: #check for numeric variable
        summary = sal[variable].describe() #summary stats
        skewness = sal[variable].skew() #skewness
        kurtosis = sal[variable].kurtosis() #kurtosis

        print("STATS: ")
        print(summary)

        print("\n")

        print("Other Stats: ")
        print(f"Skewness: {skewness}")
        print(f"Kurtosis: {kurtosis}")
    else:
        print(f"Error: The chosen variable '{variable}' is not numeric.") #if not numeric
else:
    print(f"Error: The chosen variable '{variable}' doesn't exist.") #if does not exist

Enter the Variable name: salary_in_usd
STATS:
count      2584.000000
mean      133409.200186
std       67136.837329
min        5132.000000
25%       84975.000000
50%      130000.000000
75%      175000.000000
max       450000.000000
Name: salary_in_usd, dtype: float64

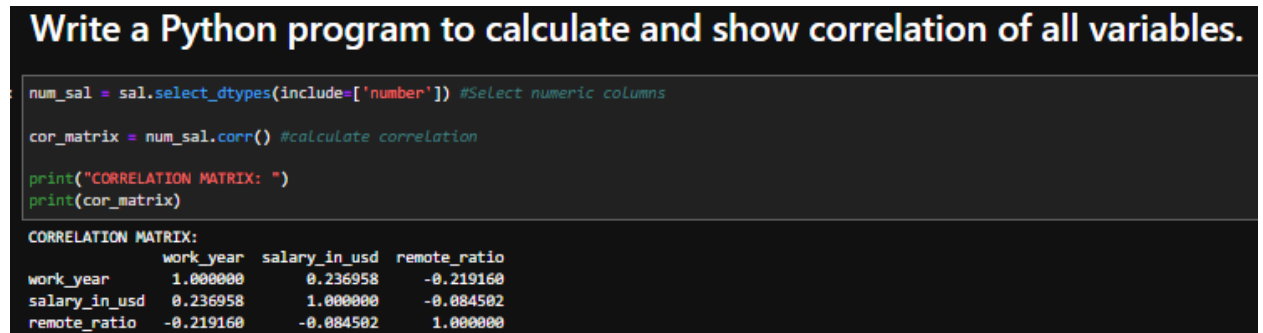
Other Stats:
Skewness: 0.6283168790580038
Kurtosis: 0.8269400876861832
```

Figure 14: Display summary stats.

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

Answer:

The '.select_dtype(include=['number'])' only selects numeric data from the dataframe. The 'num_sal.corr()' calculates the correlation matrix.



```
num_sal = sal.select_dtypes(include=['number']) #Select numeric columns
cor_matrix = num_sal.corr() #calculate correlation
print("CORRELATION MATRIX: ")
print(cor_matrix)
```

CORRELATION MATRIX:

	work_year	salary_in_usd	remote_ratio
work_year	1.000000	0.236958	-0.219160
salary_in_usd	0.236958	1.000000	-0.084502
remote_ratio	-0.219160	-0.084502	1.000000

Figure 15: calculate and show correlation of variables.

5. Data Exploration

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

Answer:

The `sal['job_title'].value_counts().head(15).index.tolist()` calculates the unique job titles from the 'job_title' columns. The `'head(15)'` method selects 15 of the most frequent jobs which is then converted into a python list after being extracted.

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

top_fifteen_jobs = sal['job_title'].value_counts().head(15).index.tolist() #Select common top 15 jobs titles
top_fifteen_jobs

['Data Engineer',
 'Data Scientist',
 'Data Analyst',
 'Machine Learning Engineer',
 'Analytics Engineer',
 'Research Scientist',
 'Data Architect',
 'Data Science Manager',
 'ML Engineer',
 'Research Engineer',
 'Applied Scientist',
 'Machine Learning Scientist',
 'Lead Data Scientist',
 'Data Science Consultant',
 'Data Manager']
```

Figure 16: Find top 15 jobs.

'Job_counts = sal['job_title'].value_counts().head(15)' counts the job frequency of the 'sal' dataframe from the column 'job_title'. The '.head(15)' chooses the most frequent top 15 job_title.

- The 'plt.figure(figsize=(12, 7))' creates a figure with the size of width 12 and the height if 7 in inches.
- Job_counts.plot(kind='bar', color='red') creates a bar graph with the color red.
- The label for the x-axis is set with the font size of 16.
- The label for the y-axis is set with the font size of 16.
- The x-label is rotated to 90 degrees.
- The 'plt.tight_layout()' prevents overlapping of the elements.

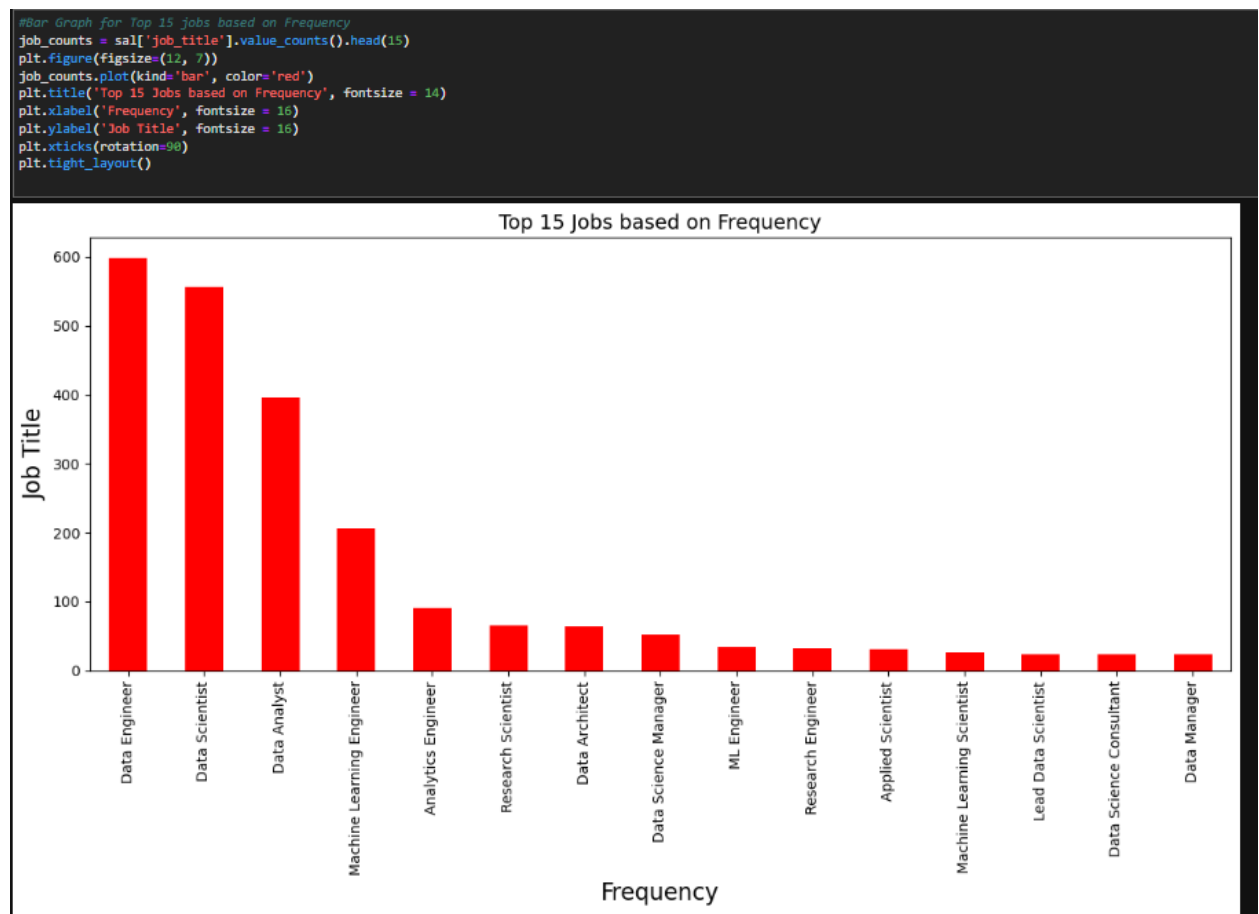


Figure 17: Display top 15 jobs in bargraph.

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

Answer:

The `.max()` method find and compares the highest salaries from 'salary_in_usd' column and stores it to 'top_salary'. The data is then located in the dataframe in which the job title with the highest salaries is found.

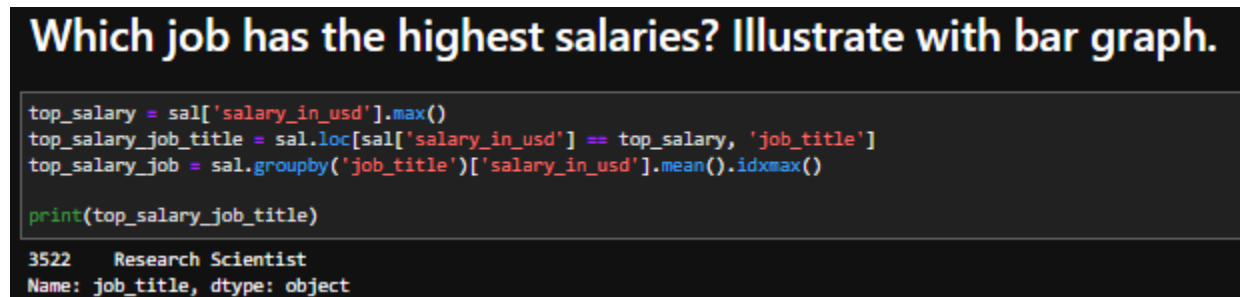


Figure 18: Job with highest salary.

- The `plt.figure(figsize=(40, 8))` creates a figure with the size of width 40 and height of 8 in inches.
- The code `'sal.groupby('job_title')['salary_in_usd'].mean().sort_values(ascending=False).plot(kind='bar', color='red')'` groups the data by job title and calculates the salary for the groups and sorts the data into descending order.
- The x-label with the font size 18 is set.
- The y-label with the font size 18 is set.
- The title for the bar graph is set with the font size 18.
- The x-label is then rotated to 90 degrees.

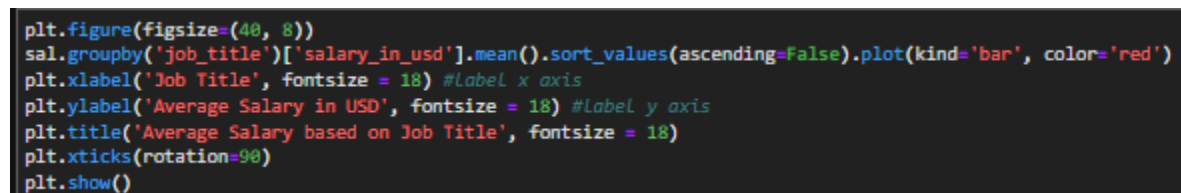


Figure 19: Display job with highest salary.

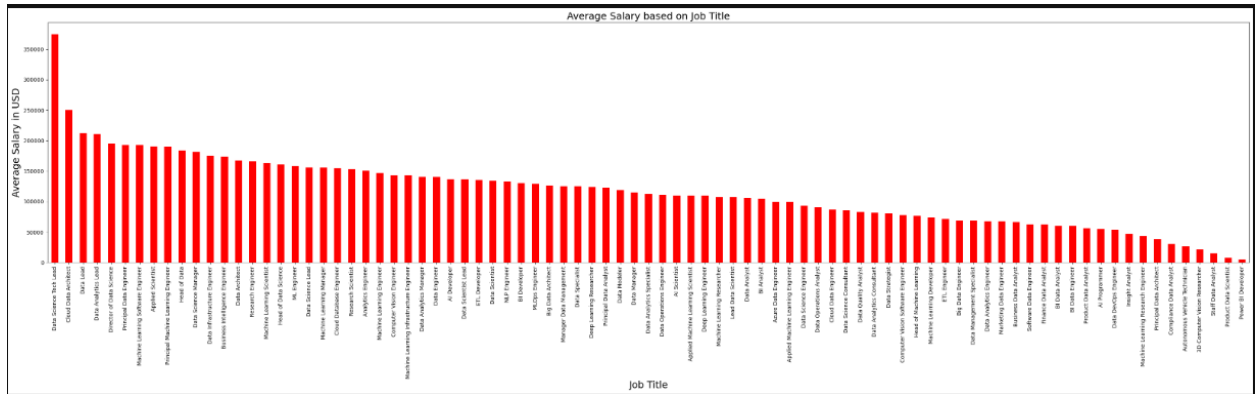


Figure 20: Display jobs with highest salary in a bargraph.

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

Answer:

The 'groupby()' method groups unique values which consists of rows with the same 'experience level'. The '.max' method calculates the maximum salary in the 'salary_in_usd' column.

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

```
salary_based_on_exp = sal.groupby('experience_level')['salary_in_usd'].max() #Group dataframe by experience_level and calculate max salary for group
salary_based_on_exp
```

```
experience_level
Entry Level      300000
Executive Level  416000
Medium Level     450000
Senior Level     423834
Name: salary_in_usd, dtype: int64
```

Figure 21: Salaries based on experience.

- The 'plt.figure(figsize=(5, 4))' creates a figure with the size of width 5 and height 4 in inches.
- The 'salary_based_on_exp.plot(kind='bar', color='red')' plots the data from the data frame 'salary_based_on_exp' and a bar graph is chosen to be displayed.
- The color is chosen, in this case red color is chosen.
- The 'plt.title' sets the title for the bargraph and the fontsize of 16 is chosen.
- The x-axis is then labeled with the font size of 14.
- The y-axis is then labeled with the font size of 14.
- A 45-degree tilt is added to the x-axis label.
- The 'plt.tight_layout()' prevents overlapping of the elements.



Figure 22: Display salaries based on experience in a bargraph.

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

Answer:

- The 'plt.figure(figsize=(14, 6))' creates a figure with the size of width 14 and height 6 in inches.
- The 'plt.subplot(1, 2, 1)' the first argument '1' sets the number of rows, second argument '2' sets the number of columns and the third argument '1' sets the index for subplot.
- An array which contains the salary of the employees in USD is plotted and bins are created to divide the data.
- A color is chosen, in this case red color is chosen and the edge color is set to white for better understanding.
- 'plt.title('Salary Distribution plotted in Histogram', fontsize=14)' creates a sub plot with the font size of 14.
- A label to the x-axis is added with the font size of 13 and a label to the y-axis is added with the font size of 13.
- The text under the x-axis is tilted to 45 degrees.

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

```
#Histogram of salary_in_usd
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.hist(sal['salary_in_usd'], bins=20, color='red', edgecolor='white')
plt.title('Salary Distribution plotted in Histogram', fontsize=14)
plt.xlabel('Salary in USD', fontsize=13)
plt.ylabel('Frequency', fontsize=13)
plt.xticks(rotation=45)

(array([-100000.,      0., 100000., 200000., 300000., 400000.,
        500000.]),
 [Text(-100000.0, 0, '-100000'),
  Text(0.0, 0, '0'),
  Text(100000.0, 0, '100000'),
  Text(200000.0, 0, '200000'),
  Text(300000.0, 0, '300000'),
  Text(400000.0, 0, '400000'),
  Text(500000.0, 0, '500000')])
```

Figure 23: Plot histogram.

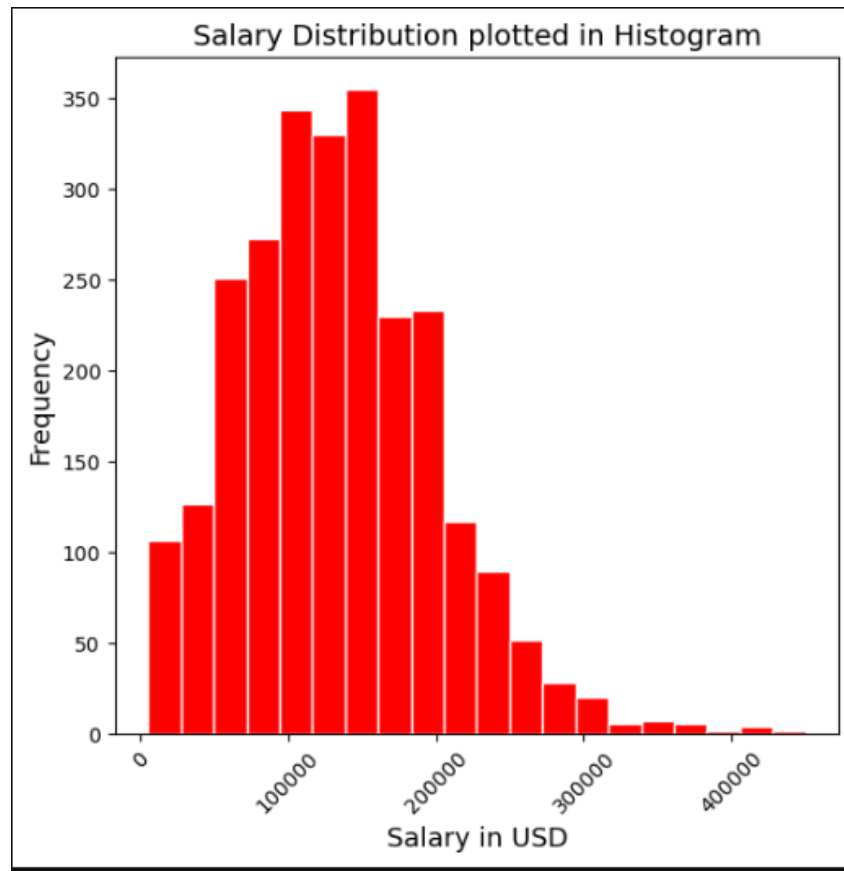


Figure 24: Display histogram.

- A boxplot of salary_in_usd is also displayed.
- 'plt.boxplot(sal['salary_in_usd'])', creates a boxplot of data from 'salary_in_usd'.
- 'notch=True, vert=True, patch_artist=True', adds notches, draws the box plot vertically and adds color inside the box.
- 'Flierprops' sets the circles with the color red and sets the border color to black. The marker is set 'o' which displays the marker as 'o' and its size is set to 5.
- 'Medianprops' sets the thickness for the median line.
- 'boxprops' fills the color with red and sets the edges to black.
- The title is then set with the font size of 16.
- The label for the y-axis is set with the font size of 12.
- A guideline is added for accurate understanding of the figure.

```
#Box plot of salary_in_usd
plt.boxplot(sal['salary_in_usd'],
            notch=True, #adding notches
            vert=True, #plot the box plot vertically
            patch_artist=True,
            flierprops={"marker": 'o', "markerfacecolor": "red", "markeredgecolor": "black", "markersize": 5},
            medianprops={"linewidth": 3},
            boxprops={"facecolor": "red", "edgecolor": "black"},
            capprops={"linewidth": 1})

plt.title('Salary Distribution (USD)', fontsize=16)
plt.ylabel('Salary in USD', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7) #Add guidelines for better understanding

plt.show()
```

Figure 25: code for displaying boxplot Display boxplot.

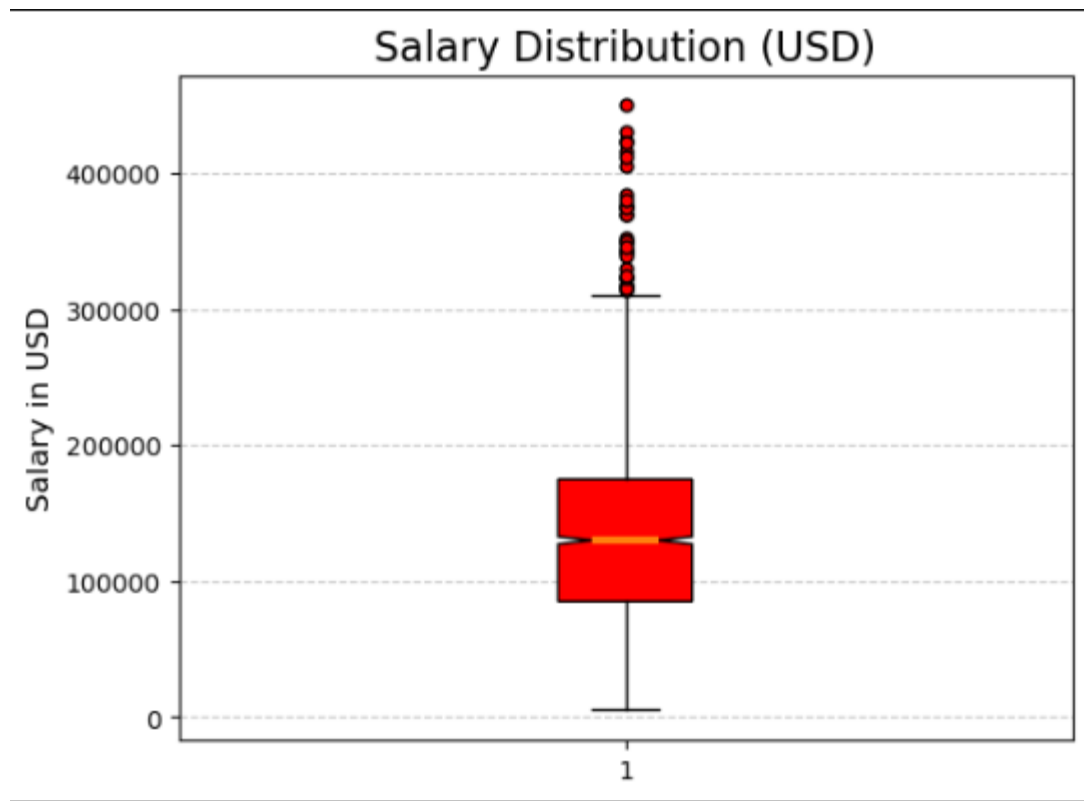


Figure 26: Display boxplot.

6. Conclusion

This coursework provides valuable knowledge in how data exploration is done. It also teaches data understanding and data cleaning. The data provided were inconsistent and unoptimized. This coursework required students to clean and optimize the data allowing for optimal data exploration. Creating statistics and visualization such as bar graphs and histograms of the data helped understand the data better. This provided with deeper understanding about the data and help identify quality issues.

In conclusion, this coursework helped in gaining deeper understanding about analysis and exploration of data. It has improved the problem-solving abilities in examining complicated datasets. The technical documentation also helps verify the analysis of the data confirming results. Successfully loading, cleaning, and exploring of data was done through Python programming language.