LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

CC5061NI

60% Individual Coursework

2023-24 Spring

Student Name: Sajin Raj Amatya

London Met ID: 22067177

College ID: NP01AI4A220003

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Monday, May 13, 2024

Word Count: 3640

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. Data Understanding 1
2. Data Preparation 4
2.1 Loading the data into the pandas Data frame. 4
2.2 Removing unnecessary columns. 5
2.3 Removing Nan missing values from the updated data frame 6
2.4 Checking for the duplicate's values. 7
2.5 Uniques value from all the column in the data frame 8
2.6 Renaming values of experience level column. 9
3. Data Analysis 10
3.1 Summary statistic of any chosen variable. 10
3.2 Correlation of all variables. 12
4. Data Exploration 14
4.1 Top 15 jobs 14
4.2 Job with the highest salary 16
4.3 Salaries based on experience level. 18
4.4 Histogram and boxplot of any chosen different variable 20
4.4.1 Histogram plot for work_year column 20
4.4.2 Boxplot for salary column 22
5. References 24

Table of Figure

Figure 1 Code: Python code to load data into pandas Data frame 4
Figure 2 Output: Displaying the first 4 rows of the data frame. 4
Figure 3 Code: Python code to Remove unnecessary column from the data frame . 5
Figure 4 Output: Columns of the data frame after removing salary and salary_currency columns. 5
Figure 5 Python code to remove Nan missing values for the updated data frame 6
Figure 6 Output: Inspection of null values. 6
Figure 7 Code: Python code to check duplicate values in the data frame 7
Figure 8 Output: Displaying the duplicates data 7
Figure 9 Code : Python code to find out the unique value from all the column 8
Figure 10 Output: Unique value for each column. 8
Figure 11 Code : Python code to rename the value of experience level column 9
Figure 12 Output: First 5 data of experience_level column after replacing it with new one 9
Figure 13 Code: Python program to summary statistics 10
Figure 14 Output: Summary statistic of salary_in_usd column 11
Figure 15 Code : Python code to calculate the correlation of all variable 12
Figure 16 Output : Correlation between of all numeric columns 13
Figure 17 Code: python code to find out top 15 jobs 14
Figure 18 Output: Top 15 jobs. 15
Figure 19 Output: Bar graph for top 15 jobs 15
Figure 20 Code : Python code to find out the job with the highest salary 16
Figure 21 Jobs with the highest salary. 17
Figure 22 Output: Bar graph for top 10 job with the highest salary. 17
Figure 23 Code : Python code to find out salaries based on experience level 18
Figure 24 Output : Salary of employee based on experience level 19
Figure 25 Output : Bar graph of salary of employee based on experience level 19
Figure 26 Code : Python code to create histogram for work_year column 20
Figure 27 Output : Histogram of work_year column 21
Figure 28 Code: Python code for creating box plot for salary in usd vs experience level 22
Figure 29 Output: Boxplot of experience level by salary in usd. 23

Table of tables

Table 1 Data Understanding 3

1. Data Understanding

S.No	Column Name	Description	Data Type
1.	work_year	In work_year column there is a total of 4 unique values for the work_year column i.e. '2023', '2022', '2021' and '2020' year where the highest number of occurrences for employee work year is "2023" year with 1785 records and the lowest number of occurrences is "2020" year with 76 records.	64 bits integer
2.	experience_level	'Experience_level 'columns indicate the level of experience of the employees for a particular work they have done. There is a total of 4 experience levels in the data frame. i.e. 'SE', 'MI',' EN', and 'EX' where among them 'The SE', experience level has the highest number of records '2516' and 'EX' experience with the lowest number of records '114'.	Object
3.	employment_type	'employment_type' columns indicate the type of employment the employee is associated with for the job they are working. There are a total of 4 employment types "FT (full-time)", "PT (part-time), "CT (Contract- term) "and "FL (freelance)". Most of the Employees i.e. 3718 out of 3755 have a full-time job.	Object

4.	job_title	'job_title' columns indicate the job title of the employee. There is a total of 93 unique job titles in the dataset. Most of the employee's job title is "Data engineer".i.e. 1040 out of 3755 records in the dataset.	Object
5.	salary	The salary of the employees based on the job title, experience level, and employment type is contained in this column. The highest salary is '30400000 CLP ' for the job title "Data Scientist" with full-time employment and mid-level experience.	64 bits integer
6.	salary_currency	The salary currency of the salary for each row is denoted by the salary_currency column. This column is removed in data preparation stage.	Object
7.	salary_in_usd	The salary_in_usd columns include the salary with their respective currency that has been converted into USD.	64 bits integer
8.	employee_residence	This column represents the home country of each employee in the data set. There are a total of 20 unique residence values in this column. Most of the employee resides in the US with a total number of residences of 3004 employees.	Object
9.	Remote_ratio	This column shows the remote ratio of the company that the employee works on. There are a total of 3 unique values 0, 50, and 100. Most of the employee companies have 0 remote ratio percent i.e. they only hire employees for On-site office work.	64 bits integer
10.	company_location	The location of the company where the employee works is represented by this column. There are a total of 72 locations in this column. Among them, most of the company is in 'US' i.e. 3040 companies.	Object

11.	company_size	The size of the company is represented by this column. There are a total of three size companies as per the data, they are large(l), medium(M), and small (S) size companies. Most of the companies are of Large size i.e. 3153 companies.	Object

Table 1 Data Understanding

2. Data Preparation

2.1 Loading the data into the pandas Data frame.

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

Pandas, a library of Python is used to load the CSV data into the data frame using the .read_csv() function. To use the read_csv() function, we have to import the pandas library with proper aliasing. In the function, the correct CSV file path of the dataset is given as the parameter. Then, it reads the CSV files and converts them into the data frame and stores it in the variable name 'salaries_dataframe'.

We have to inspect if the data set is loaded to the variable using the .head() function which displays the desired number of rows of the dataset as per the number passed as a parameter in it

Code:

```
[57]:
import pandas as pd
#loading the csv file into pandas data frame using pd.read_csv() method
salaries_dataframe = pd.read_csv("DataScienceSalaries.csv")
#Display first four rows of the dataframe
salaries_dataframe.head(4)
```

Figure 1 Code : Python code to load data into pandas Data frame

Output:

The first 4 rows of the data frame are displayed after data is successfully loaded in the data frame.

[57]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	100
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	100
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	100

Figure 2 Output: Displaying the first 4 rows of the data frame.

2.2 Removing unnecessary columns.

Q.N Write a Python program to remove unnecessary columns i.e., **salary and salary currency**.

Removing unnecessary columns from the data frame helps to reduce memory and resource division for faster and optimized output while conducting data analysis (Wizards, 2023).

The function .drop() of pandas, the Python library is used in the code below to remove the column salary and salary currency from the data frame 'salaries_dataframe'. In the parameter of the function, the column name to be removed from the data frame is passed, the axis is set to 1 which indicates to drop the column, place is set to true which will delete the column of the original data frame instead of creating a new one (codecademy, 2022).

Code :

```
]: # removing the two column 'salary' and salary_currency from the data set
   # using drop method from the pandas with the parameter columns name, axis and inplace
   salaries_dataframe.drop(['salary', 'salary_currency'], axis=1, inplace=True)
```

Figure 3 Code: Python code to Remove unnecessary column from the data frame

Output:

Displaying all the columns after removing salary and salary_currency from the data frame.

```
: # Columns of the dataframe after removing the two columns
  print(salaries_dataframe.columns)

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

Figure 4 Output: Columns of the data frame after removing salary and salary_currency columns.

2.3 Removing Nan missing values from the updated data frame

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

Code:

```
# Removing the Missing values from the data set
salaries_dataframe.dropna(axis = 0 , how = "all",inplace=True)
```

Figure 5 Python code to remove Nan missing values for the updated data frame

Missing values can misinterpret the insight generated during the data analysis process (Masterindatascience, 2024). We must either remove the row with the missing value or perform the correct imputation.

The dropna() function is used to remove the missing values for the data frame. Its parameter is ‘axis’ for checking the missing values either for the row ‘0’ or column ‘1’, ‘how’ parameter to remove a row or column if it has at least one missing value ‘any’ and “all” if it has all the values missing In a row or column, ‘inplace’ parameter which will either drop the original data frame if set to True or create a new data frame if set to False. (MachineLearningPlus, 2024).

Output :

```
# inspection of null values
salaries_dataframe.isna().sum()
```

```
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
```

Figure 6 Output: Inspection of null values.

There are no ‘Nan” values in the data frame as per the inspection of null values.

2.4 Checking for the duplicate's values.

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

Code:

```
duplicates_value = salaries_dataframe[salaries_dataframe.duplicated(keep=False)]
duplicates_value
```

Figure 7 Code: Python code to check duplicate values in the data frame

Inspection of the duplicated values is performed using .duplicated() function of pandas which takes 1 parameter with a value false in the above code. The keep parameter with the value False sets all the duplicate values to be true. When the keep parameter is set to first then it doesn't set the first occurrence duplicate to True. if the keep parameter is set to last then it doesn't set the last occurrence duplicate to True.

Output

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
5	2023	Senior Level/Expert	FT	Applied Scientist	222200	US	0	US	L
6	2023	Senior Level/Expert	FT	Applied Scientist	136000	US	0	US	L
9	2023	Senior Level/Expert	FT	Data Scientist	147100	US	0	US	M
10	2023	Senior Level/Expert	FT	Data Scientist	90700	US	0	US	M
11	2023	Senior Level/Expert	FT	Data Analyst	130000	US	100	US	M
...	...	...	...	...	...	...	...	...	...
3441	2022	Senior Level/Expert	FT	Data Engineer	115000	US	100	US	M
3502	2021	Medium Level/Intermediate	FT	Data Engineer	200000	US	100	US	L
3586	2021	Medium Level/Intermediate	FT	Data Engineer	200000	US	100	US	L
3665	2021	Medium Level/Intermediate	FT	Data Scientist	90734	DE	50	DE	L
3709	2021	Medium Level/Intermediate	FT	Data Scientist	90734	DE	50	DE	L

1715 rows × 9 columns

Figure 8 Output: Displaying the duplicates data

There are a total of 1715 duplicate values including all in the data frame. There might be employees of the same salary, job title, experience level, etc. I think it doesn't affect our data analysis because employees with similar job descriptions are possible.

2.5 Unique value from all the column in the data frame

Code:

```
# storing the name of the column in the list
column_list = list(salaries_dataframe.columns)
# iterating through each column
for column in column_list:
    print("")
    print("Unique value of:",column +" "+ "column")
    print(salaries_dataframe[column].unique())
```

Figure 9 Code : Python code to find out the unique value from all the column

The name of the column is stored in the list name “column_list”. The for each loop is used to iterate over each column and execute .unique() function to find out the unique value associated with that column.

Output:

```
Unique value of: work_year column
[2023 2022 2020 2021]

Unique value of: experience_level column
['Senior Level/Expert' 'Medium Level/Intermediate' 'Entry Level'
 'Executive Level']

Unique value of: employment_type column
['FT' 'CT' 'FL' 'PT']

Unique value of: job_title column
['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']

Unique value of: salary_in_usd column
[ 85847  30000  25500 ...  28369 412000  94665]

Unique value of: employee_residence column
['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 value of: remote_ratio column
[100   0  50]

Unique value of: company_location column
['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 value of: company_size column
['L' 'S' 'M']
```

Figure 10 Output: Unique value for each column.

2.6 Renaming values of experience level column.

Code :

```
# creating a dictionary where the current column values as key  to new columns as values of dictionary for experience_level column
dict_rename = {"SE":"Senior Level/Expert","MI":"Medium Level/Intermediate", "EN" : "Entry Level", "EX" : "Executive Level"}
# renaming the value by using the above defined dict_rename
salaries_dataframe['experience_level'] = salaries_dataframe.experience_level.replace(dict_rename)
print(salaries_dataframe.experience_level.head(5))
```

Figure 11 Code : Python code to rename the value of experience level column

To rename the column value of the data frame. replace() function is used in the above code which takes the dictionary as a parameter. The dictionary is created where all the key is the name of the current column values and values for the dictionary are the names of the all-new column values.

Output :

```
0          Senior Level/Expert
1    Medium Level/Intermediate
2    Medium Level/Intermediate
3          Senior Level/Expert
4          Senior Level/Expert
Name: experience_level, dtype: object
```

Figure 12 Output: First 5 data of experience_level column after replacing it with new one

Display in the first 5 data of experience_level column after replacing the values.

3. Data Analysis

3.1 Summary statistic of any chosen variable.

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

Code :

```
salaries_sum = salaries_dataframe['salary_in_usd'].sum()
salaries_mean = salaries_dataframe['salary_in_usd'].mean()
salaries_std = salaries_dataframe['salary_in_usd'].std()
salaries_skewness = salaries_dataframe['salary_in_usd'].skew()
salaries_kurtosis = salaries_dataframe['salary_in_usd'].kurtosis()
print("Summary statistic of salary_in_usd column " )
print("Sum of salary column:", salaries_sum)
print("Mean of salary column :", salaries_mean)
print("Standard deviation of salary column:", salaries_std)
print("Skewness of salary column :", salaries_skewness)
print("Kurtosis of salary column :", salaries_kurtosis)
```

Figure 13 Code: Python program to summary statistics

The summary statistics of sum, mean, standard deviation,skewness and kurtosis show for salary_in_usd column of the data frame using .sum() function for sum value, .mean() for mean value , .std() for standard deviation value, .skew() for skewness value, and .kurtosis() for kurtosis values of salary_in_usd column.

Output :

```
Summary statistic of salary_in_usd column
Sum of salary column: 516576814
Mean of salary column : 137570.38988015978
Standard deviation of salary column: 63055.625278224084
Skewness of salary column : 0.5364011659712974
Kurtosis of salary column : 0.8340064594833612
```

Figure 14 Output: Summary statistic of salary_in_usd column

- The Sum of all the values of the salary_in_usd column is 516576814
- The average value of salary_in_usd is 137570.389
- The standard deviation measures the variability in the dataset which indicates to us how far the reach values of lie from the mean (Bhandari, 2020). The standard deviation of 63055 tells us that data of the salary column differ from the mean of 137570.389 by around on average 63055.
- Skewness is the measure of the degree of asymmetry of a distribution (Turney, 2022). The skewness of salary_in_usd column is 0.53 > 0 so the distribution is positive skew which indicates that most people earn low to moderate salaries, but many fewer employees earn the highest salary who are on executive level.
- Kurtosis of 0.83 denotes that the tail of the distribution of the salary in usd column data is longer compared to the normal distribution which tells us that there is a lot of outliers in the salary in usd column.

3.2 Correlation of all variables.

Q.n) Write a Python program to calculate and show the correlation of all variables.

Code:

```
# calculating the correlation of all the column with numeric values
correlation_all_column = salaries_dataframe[['salary_in_usd','remote_ratio','work_year']].corr()
print(correlation_all_column)
```

Figure 15 Code : Python code to calculate the correlation of all variable

Correlation is the association between the two variables of the data. It shows the strength between the variables. There are three types of correlation, positive and negative, and no correlation. If there is a positive correlation between two variables, then the increased value in the independent variable also increases the value of the dependent variable. In case of a negative correlation, the increase in the independent variable decreases the values of the dependent variable. No correlation denotes there is no relationship between two variables in a dataset.

The .corr() function is used to calculate the correlation between the variable or column in the data frame. In the code above, the column with numeric data type is selected i.e. ‘salary_in_usd’, ‘remote_ratio’, and ‘work_year’ from the data frame, and .corr() function is used to calculate the correlation among these three numeric columns.

Output :

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

Figure 16 Output : Correlation between of all numeric columns

- There is a negative correlation between ‘remote_ratio’ and ‘salary_in_usd’ column which indicates that On-site office work pays more salary than remote work in the company.

- There is a positive correlation between “work_year” and “Salary_in_usd” which indicates that every year the salary of the employee increases by a certain amount.

- There is a negative correlation between the ‘remote_ratio’ and “work_year ” columns which indicates that every year the remote job is decreasing.

4. Data Exploration

4.1 Top 15 jobs

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

Code:

```
]: #calculating the top 15 jobs from the data set
top_15_jobs = salaries_dataframe.job_title.value_counts().nlargest(15)
# Displaying out the top 15 job
print(top_15_jobs)

# Figure size of the plot
plt.figure(figsize=(8, 5))
# plotting bar plot
plt.bar(top_15_jobs.index,top_15_jobs.values,color='green')
# title of the plot
plt.title('Top 15 jobs ')
#x label name of the plot
plt.xlabel('Job Title')
# y label name of the plot
plt.ylabel('Number of jobs')
# rotating the x axis tick to 90 degree for clear visibility
plt.xticks(rotation=90)
# setting the range of the y label tick values
plt.yticks(range(0, 1200, 100))
# saving the figure
plt.savefig('top15.png')
plt.show()
```

Figure 17 Code: python code to find out top 15 jobs

To find out the top 15 jobs for the data set, the ‘job_title’ column of the salaries_dataframe was selected. The .value_count() function along with .nlargest was used to extract the top 15 jobs. .value_count() function would count the occurrence of all the unique values of the job_title columns and .nlargest() function with parameter 15 would return the top 15 values with the highest count for the job_title column. Then the result was stored in the top_15_job variable.

Bar plot was plotted to show the top 15 jobs using .bar() function of matplotlib that takes 3 parameters in the above code, x is an index of the top_15_job variable, height is the values of top_15_job variable, and colour is set to green. Proper title, x-axis, and y-axis label are also given with appropriate names. The x-axis ticks were rotated to 90 degrees for better visibility and the value of the y-axis ticks count was set to the range of 0 to 1200 with step size 100. At last, the figure was saved with the name “top15.png”.

Output :

```
job_title
Data Engineer                1040
Data Scientist                840
Data Analyst                  612
Machine Learning Engineer     289
Analytics Engineer            103
Data Architect                101
Research Scientist             82
Data Science Manager           58
Applied Scientist              58
Research Engineer              37
ML Engineer                    34
Data Manager                   29
Machine Learning Scientist     26
Data Science Consultant        24
Data Analytics Manager         22
Name: count, dtype: int64
```

Figure 18 Output: Top 15 jobs.

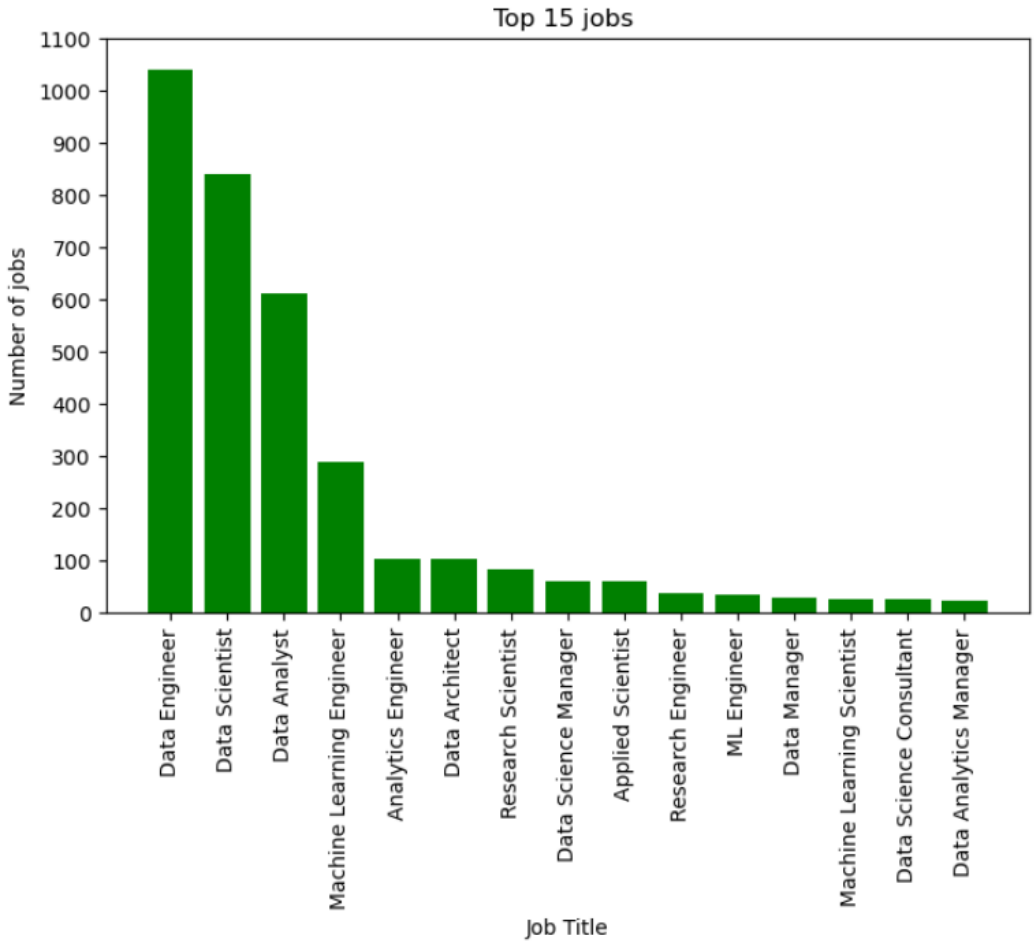

Figure 19 Output: Bar graph for top 15 jobs

From the bar plot, the highest number of jobs is ‘Data Engineer’ with 1040 jobs.

4.2 Job with the highest salary

Which job has the highest salaries? Illustrate with bar graph

Code:

```
[7]: #finding out the job  with the highest salary
highest_salary = salaries_dataframe.groupby('job_title')['salary_in_usd'].mean().nlargest(1)
print("The job with highest salary :")
print(highest_salary)
# Top 10 jobs with highest salary
Top_10_highest_salary = salaries_dataframe.groupby('job_title')['salary_in_usd'].mean().nlargest(10)
# figure size
plt.figure(figsize=(8, 5))
# bar plot
plt.bar(Top_10_highest_salary.index,Top_10_highest_salary.values, color='purple')
# title of the bar plot
plt.title('top 10 jobs with highest salary')
# name of x label
plt.xlabel('Job Title')
# name of y label
plt.ylabel('Salary (USD)')
# rotating the x axis tick to 90 degree
plt.xticks(rotation=90,)
# saving the figure
plt.savefig("top_10_job_with_highest_salary.png")
plt.show()
```

Figure 20 Code : Python code to find out the job with the highest salary

The function. groupby() of pandas is used to group rows with a similar value of the column to perform an aggregate function based on a specified column (Ravikiran, 2023).

In the above code, salaries dataframe is grouped based on values of the job_title column, after grouping the data in “salary_in_usd” column aggregate function .mean() is performed along with nlargest() function with 1 parameter which finds out the job with the highest salary.

Bar plot is used to plot the top 10 jobs with the highest salary by using .bar() function with the proper title of the plot using plt.title(), x-axis and y-axis label name using plt.xlabel() and plt.ylabel(), rotation of the x-axis tick for better visibility of the jobs indexes. At last, the bar plot is with the file name “top_10_jobs_with_highest_salary.png” saved using plt.savefig() function

Output:

```
The job with highest salary :
job_title
Data Science Tech Lead    375000.0
```

Figure 21 Jobs with the highest salary.

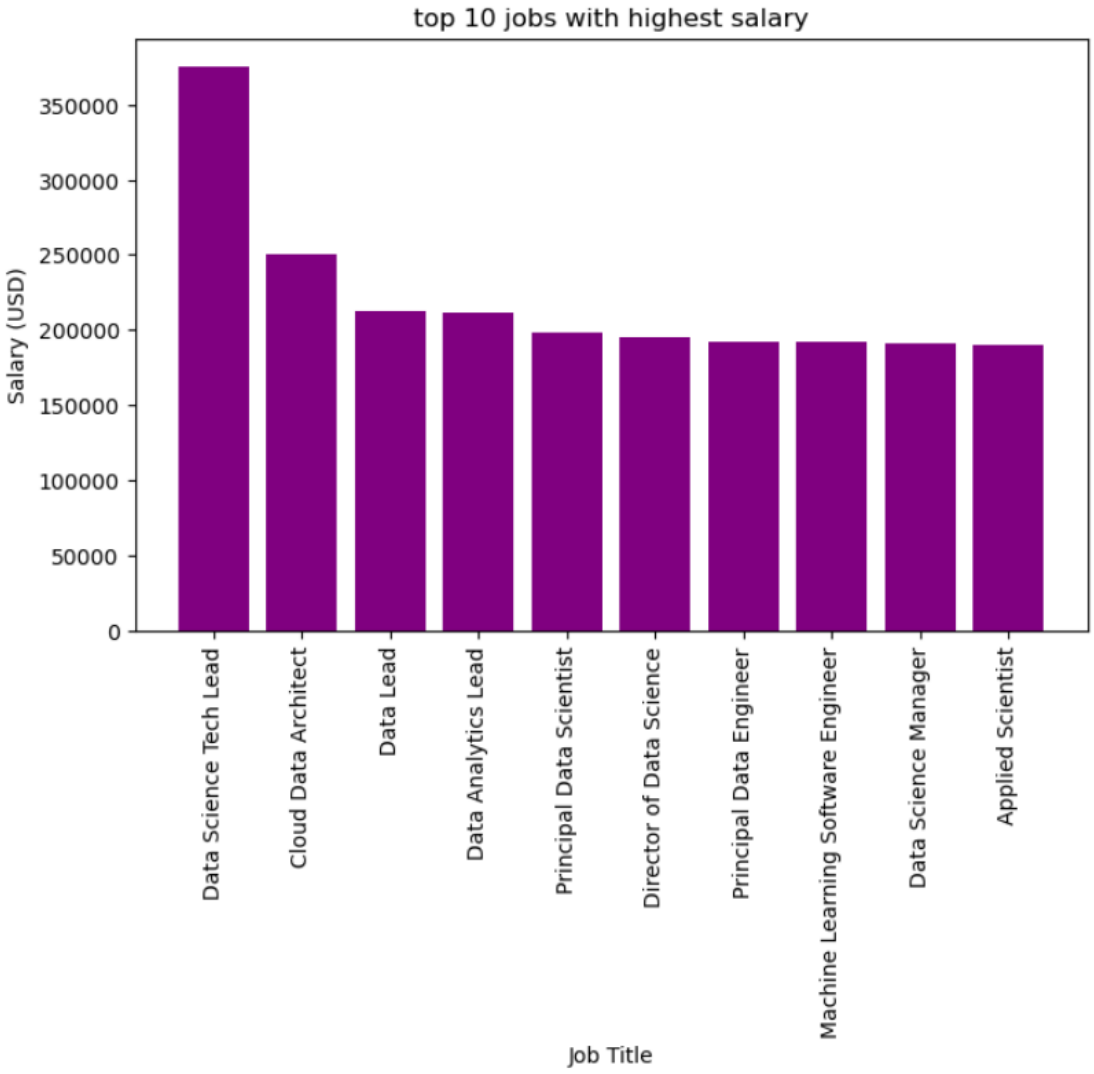

Figure 22 Output: Bar graph for top 10 job with the highest salary.

From the bar plot, the highest-paying job is ‘Data Science Tech lead ‘.

4.3 Salaries based on experience level.

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

Code:

```
]: # calculating the salary based on the experience_level
salary_based_on_experience_level = salaries_dataframe.groupby('experience_level')['salary_in_usd'].mean()
print("Salary based on thhe experience_level: " )
print(salary_based_on_experience_level)

# Size of the figure
plt.figure(figsize=(10, 6))
# plotting bar graph
plt.bar(salary_based_on_experience_level.index,salary_based_on_experience_level.values, color ="orange")
# title of the plot
plt.title('Salary of the employees based on the experience level')
# x label of the plot
plt.xlabel('Experience Level ')
# y label of the plot
plt.ylabel('Salary (USD)')
# x axis tick rotation
plt.xticks(rotation=10)
# saving the figure
plt.savefig("experience.png")
plt.show()
```

Figure 23 Code : Python code to find out salaries based on experience level

To find out the salaries based on experience level, group by function of pandas is used to group the similar values of the experience_level column of the data frame. Then, the column salary_in_usd is selected to perform an aggregate function for calculating the salary based on the experience level.

Illustration of the data through a bar graph is done by using .bar function of matplotlib which takes three parameter values for the x-axis , y-axis, and colour of the bar plot. The proper title, x, and y label is added to the plot using plt.title(), plt.xlabel() and plt.ylabel() functions of matplotlib. At last, the plot is saved with the file name “experience.png” using plt.savefig() function.

Output:

```
Salary based on thhe experience_level:
experience_level
Entry Level                      78546.284375
Executive Level                 194930.929825
Medium Level/Intermediate       104525.939130
Senior Level/Expert             153051.071542
```

Figure 24 Output : Salary of employee based on experience level

Figure 25 Output : Bar graph of salary of employee based on experience level

Employees with executive level experience level have more salary compared to other experience. The entry-level experienced employee has the lowest salary compared to other experiences. Therefore, the employee's salary is increases if they have more work experience.

4.4 Histogram and boxplot of any chosen different variable

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

4.4.1 Histogram plot for work_year column

Code :

```
# calculating the desire number of bins for the histogram
binss = salaries_dataframe['work_year'].max() - salaries_dataframe['work_year'].min()
# fiugure size
plt.figure(figsize=(6, 4))
# plotting histogram for work_year columm
plt.hist(salaries_dataframe['work_year'], bins =binss,color = "green" )
# x axis  ticks values
plt.xticks(salaries_dataframe['work_year'].unique())
# y axis ticks value
plt.yticks(range(0,4500,400))
# title of the plot
plt.title("frequency distribution of work_year column " )
# name of the xlabel
plt.xlabel('work year')
# name of the y label
plt.ylabel("frequency")
# include grid in the histogram
plt.grid()
plt.savefig("hist.png")
plt.show()
```

Figure 26 Code : Python code to create histogram for work_year column

The graphical illustration of the frequency distribution of the data is known as histogram (Byju's, 2024) .

To plot the histogram of the work_year column of the data frame, .hist() function from matplotlib is used where 3 parameters, columns bins and colour in the above code. The bins of the histogram is determined by difference of maximum and minimum value of the column for equal bin distribution and easy interpretation of the data. Ther value of x- tick is set to unique values of the work_year column and value of y-tick is set the interval of 0 to 4500 with 400 steps. The proper title, x, and y label is added to the plot using plt.title(), plt.xlabel() and plt.ylabel() functions of matplotlib. The grid line is shown in the plot by using .grid() function. At last, the plot is saved with the file name “hist.png” using plt.savefig() function

Output :

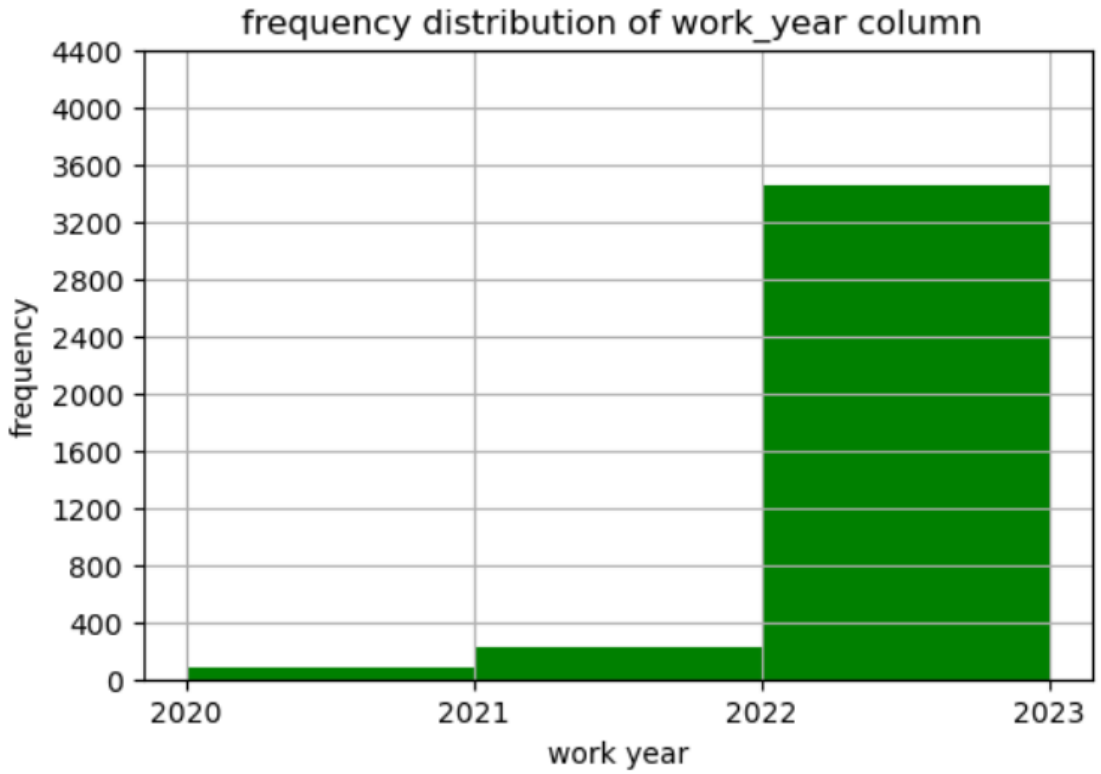

Figure 27 Output : Histogram of work_year column

From the histogram, there are high frequency count of the work year from 2022 to 2023 and low frequency count from 2020 to 2022. Most of the data is concentrated between 2022 and 2023 year.

4.4.2 Boxplot for salary column

Code :

```
# figure size
plt.figure(figsize=(10, 6))

# using seaborn library for plotting boxplot
sns.boxplot(x =salaries_dataframe['experience_level'], y = salaries_dataframe['salary_in_usd'])
# title of the plot
plt.title("Box plot of  experience level by salary in usd" )
# name of the xlabel
plt.xlabel('experience level')
# name of the y label
plt.ylabel("Salary (USD)")
plt. savefig("box.png")
plt.show()
```

Figure 28 Code: Python code for creating box plot for salary in usd vs experience level

Boxplot is a type of plot which displays the distribution of the data based on five summary statistics i.e. maximum, minimum, median, first quartile, and third quartile (Whitfield, 2023).

The function. Boxplot() from seaborn library is used in the above code to plot the box plot for salary_in_usd column of the data set. This function takes two parameter x axis values and y axis values in the above code. The proper title, x, and y label is added to the plot using plt.title(), plt.xlabel() and plt.ylabel() functions of matplotlib. At last, the plot is saved with the file name “box.png” using plt.savefig() function

Output:

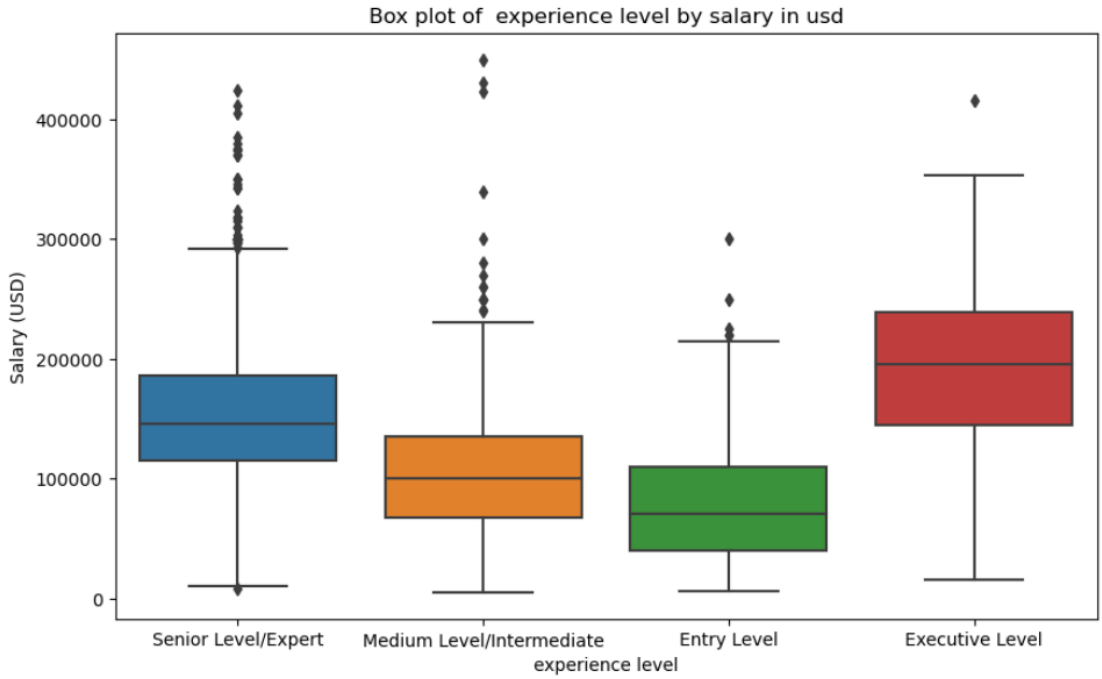

Figure 29 Output: Boxplot of experience level by salary in usd.

In the box plot, the whole box represents the interquartile range, 50 percent of the whole data, the line in the middle of the box is the median value, the lower whisker end line denotes the minimum value and the upper whisker end line denotes the maximum value. The data outside the whisker represented by the dot is outliers. There are a lot of outliers in senior level/expert salary and need to be removed for accurate results. The executive level employees have the highest salary due to the long whisker in the upper sides.

5. References

Bhandari, P., 2020. *How to Calculate Standard Deviation (Guide) | Calculator & Examples.* [Online]
Available at: https://www.scribbr.com/statistics/standard-deviation/
[Accessed 10 May 2024].

Byju's, 2024. *Histogram.* [Online]
Available at: https://byjus.com/maths/histogram/
[Accessed 11 May 2024].

codecademy, 2022. *pandas data frame . drop().* [Online]
Available at: https://www.codecademy.com/resources/docs/pandas/dataframe/drop
[Accessed 10 May 2024].

MachineLearningPlus, 2024 . *Pandas Dropna – How to drop missing values?.* [Online]
Available at: https://www.machinelearningplus.com/pandas/pandas-dropna-how-to-drop-missing-values/
[Accessed 10 May 2024].

Masterindatascience, 2024. *How to Deal with Missing Data.* [Online]
Available at: https://www.mastersindatascience.org/learning/how-to-deal-with-missing-data/
[Accessed 10 May 2024].

Ravikiran, 2023. *How to Aggregate Data Using Group By in SQL.* [Online]
Available at: https://www.simplilearn.com/tutorials/sql-tutorial/group-by-in-sql#:~:text=The%20Group%20By%20statement%20is,the%20SELECT%20command%20in%20SQL.
[Accessed 11 May 2024].

Turney, S., 2022. *Skewness | Definition, Examples & Formula.* [Online]
Available at: https://www.scribbr.com/statistics/skewness/
[Accessed 10 May 2024].

Whitfield, B., 2023 . *Understanding Boxplots.* [Online]
Available at: https://builtin.com/data-science/boxplot
[Accessed 12 May 2024].

Wizards, D. S., 2023. *Beginners Guide to Feature Selection.* [Online]
Available at: https://medium.com/@datasciencewizards/beginners-guide-to-feature-selection-ba8583b97606
[Accessed 10 May 2024].