LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

# CC5067NI-Smart Data Discovery

## 60% Individual Coursework

## 2023-24 Spring

**Student Name: Yasna Dongol**

**London Met ID: 22068112**

**College ID: Np01cp220449**

**Assignment Due Date: Monday, May 13, 2024**

**Assignment Submission Date: Saturday, May 11, 2024**

**Word Count: 2988**

*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

## Table of Figures



## Table of Tables

# 1. Data Understanding

Dataset is a lightweight collection of data stored and manipulated using different tools and libraries, which is an essential backbone for all the operations, techniques used by developers to interpret them. A dataset can be table of data with various rows and columns representing a variable or feature (geeksforgeeks, 2023). The dataset is all about the salaries in the field of Data Science. The row in the table represents a single data point whereas the columns contain different attributes related to the job such as employment type, work experience, job title, salary, currency, location.

The csv files are merged into a single file where null values are removed and updated. To retrieve the details about the dataset, '.info()' function is called, after which the details about each column of dataset is shown. The dataset contains 4 int type values and 7 object type values with 3755 non-null values. The memory usage of dataset is 322.8+KB that is almost 323 KB.

```
In [5]: annualsalary.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3755 entries, 0 to 3754
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   work_year           3755 non-null   int64 
 1   experience_level    3755 non-null   object
 2   employment_type     3755 non-null   object
 3   job_title           3755 non-null   object
 4   salary              3755 non-null   int64 
 5   salary_currency     3755 non-null   object
 6   salary_in_usd       3755 non-null   int64 
 7   employee_residence  3755 non-null   object
 8   remote_ratio        3755 non-null   int64 
 9   company_location    3755 non-null   object
 10  company_size        3755 non-null   object
dtypes: int64(4), object(7)
memory usage: 322.8+ KB
```

*Figure 1screenshot of all the information of the data set*

The following table contains the information about each column of the dataset:

*Table 1 Description of the Data Set*

| SN | Column Name | Description | Data type |
|---|---|---|---|
| 1. | work_year | This column stores the year during which the salary was paid to the employee | int |
| 2. | experience_level | This column stores the level of experience of the individual in the company | object |
| 3. | employment_type | This column stores the type of employment. | object |
| 4. | job_title | This column stores the role of job | object |
| 5. | salary | This column stores the gross salary of each job in specified currency | int |
| 6. | salary_currency | This column stores the currency of the salary paid. | object |
| 7. | salary_in_usd | This column stores the salary in USD | int |
| 8. | employee_residence | This column stores the residence of the employees in the company during the work year. | object |

| | | | |
|---|---|---|---|
| 9. | remote_ratio | This column stores the percentage of remote work done in the company | int |
| 10. | company_location | This column stores the location of the company. | object |
| 11. | company_size | This column stores the average number of people worked in the company | object |

The columns work_year consists of years from 2023-2020. Similarly, the column of depicts the experience level in the job during the year with the following possible values EN: Entry-level/ Junior, MI: Mid-level/ Intermediate, SE: Senior-level/ Expert, EX: Executive-level/ Director. The employment_type id divided into PT: Part-Time, FT: Full-Time, CT: Contract, FL: Freelance. Job_title column shows the role of an individual during the year. Salary_currency is the amount of gross salary amount paid. Salary paid in USD is represented by salary_in_usd. The primary country residence of individual during the work year is shown in column employee_residence. However, the overall amount of work done remotely is represented in remote_ratio. The average number of people that worked in the company during the year is represented by company_size, similarly the country of employer's main office or contracting branch is represented by company_location.

# 2. Data Preparation

Data preparation is the process of cleaning and transforming the raw data (structured and unstructured) to processing and analysis. Data preparation is an important step for processing and often for reformatting data, correcting data and for enriching data by combining data sets. It also involves merging data from all sources to be studied in consistent format (talend , 2024).

## 2.1 Question 1

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

Solution: To load data into the pandas dataframe we use ‘annualsalary = pd.read_csv(“DataSciencesalaries.csv”) this line read the contents of a (Comma separated values) CSV file named as “DataSciencesalaries.csv” into the dataframe of pandas designated as ‘annualsalary’. The pandas function ‘pd.read_csv’ is to read the data from the csv file and to create dataframe object, but these path of “DataSciencesalaries.csv” should be replaced with the path of csv file if it is located in different directory. The ‘annualsalary’ line simply describe the dataframe. This line is simply used to display the contents of the Dataframe.

**Loading data into pandas DataFrame**

```
In [59]: annualsalary=pd.read_csv("DataSciencesalaries.csv")

In [60]: annualsalary
```

Out[60]:

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

3755 rows × 11 columns

*Figure 2 Screenshot of loading data into pandas DataFrame*

## 2.2 Question 2

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

Solution: In python the common method to remove unnecessary columns or row we used an common method from a DataFrame by using ‘drop()’ method. According to the axis referred to the method we remove row or column (0 is set for the row and 1 is set for the column). In our python program above the ‘annualsalary.drop()’ method is used to remove the labels from column. ‘[“salary”, “salary_currency”] is the two columns containing the column names to drop. The ‘axis=1’ specifies to drop the column.

**Removing columns salary and salary_currency**

```
In [31]: annualsalary=annualsalary.drop(["salary","salary_currency"],axis=1)

In [32]: annualsalary
```

Out[32]:

| | 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 3 Screenshot of removing columns salary and salary_currency.*

## 2.3 Question 3

Question: Write a python program to remove NaN missing values from updated dataframe.

Solution: Removing Nan missing value from the dataframe is necessary to ensure the quality and correctness of data analysis. Due to the presence of null values in the dataset in the function, the function shows error while running. ‘dropna()’ function is used to remove null values from the dataframe . Using these function we can remove rows and columns containing null values, but by default this function remove NaN value containing in row.

**Removing NaN missing values**

```
In [48]: null_values = annualsalary.isna().any(axis=1)
         null_values
```

```
Out[48]: 0       False
         1       False
         2       False
         3       False
         4       False
                 ...
         3750    False
         3751    False
         3752    False
         3753    False
         3754    False
         Length: 3755, dtype: bool
```

```
In [49]: annualsalary.dropna()
```

Out[49]:

| | 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 4 Screenshot of removing NaN missing values.*

## 2.4 Question 4

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

Solution: To check duplicate values in DataFrame we use the ‘duplicated()’ method along with the method ‘any()’. The pandas dataframe ‘annualsalary.duplicated()’method is used to check the duplication of row. If the corresponding row is same as the previous row then it returns a boolean series. If the value of series is “True” then row is duplicated otherwise it is false.‘annualsalary[annualsalary.duplicated()]’ also called as an boolean indexing. According to the boolean series obtained from ‘annualsalary.duplicated()’ it filters the Dataframe ‘annualsalary’. Only the boolen series having value ‘True’ is selected. When we run these command we will get only rows that are duplicate of previous row im ‘annualsalary’ Dataframe.

**Checking Duplicate values in DataFrame**

```
In [41]: annualsalary.duplicated()
Out[41]: 0       False
         1        True
         2        True
         3        True
         4        True
                 ...
         3750     True
         3751     True
         3752     True
         3753     True
         3754     True
         Length: 3755, dtype: bool
```

```
In [7]: annualsalary[annualsalary.duplicated()]
```

Out[7]:

| | 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 Screenshot of checking duplicate values in DataFrame.*

## 2.5 Question 5

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

Solution: The line ‘for column in annualsalary.columns:’ is used to loop on each column in the DataFrame. ‘annualsalary.columns’ returnsan index object containing the column labels of the DataFrame. Within each iteration of the for loop, this line ‘unique_values = annualsalary[columns].unique() extract the value from the current column which is being iterated over annualsalary column. To return the unique value present in the column ‘unique()’ methos is used which is then assigned to the ‘unique_values’ variable.

**Displaying unique values from all the columns**

```
In [61]: for column in annualsalary.columns:
             unique_values = annualsalary[column].unique()
             print(f"The unique values in {column} are: {unique_values}")
```

```
The unique values in work_year are: [2023 2022 2020 2021]
The unique values in experience_level are: ['SE' 'MI' 'EN' 'EX']
The unique values in employment_type are: ['FT' 'CT' 'FL' 'PT']
The unique values in job_title are: ['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']
The unique values in salary are: [   80000    30000    25500   175000   120000   222200   136000   219000
   141000   147100    90700   130000   100000   213660   130760   170000
```

*Figure 6 Screenshot of displaying unique values from all the columns.*

## 2.6 Question 6

Question: Rename the experience level column as below.

a) SE – Senior Level/Expert
b) MI – Medium Level/Intermediate
c) EN – Entry Level
d) Ex – Executive Level

Solution: The above program uses the for loop statement where loop iterates over each row in the DataFrame ‘annualsalary’. ‘index’ is the index label of the row, and ‘row’ is a series containing the data in the row. The line ‘if row[experience_level”] == “SE”:’

checks whether the value in “experience_level” column of the current row is equal to “SE”. ‘annualsalary.at[index, “experience_level”] = “Senior Level/Expert”’ this program checks the condition of previous line is true or not, It also updates the value into “experience_level” column of the current row to “Senior Level/Expert”. The ‘elif’ statement also follows the same pattern as if, checking other experience levels such as MI, EN, EX and updating the values accordingly.

**Renaming the values of column experience level**

In [62]:
```
for index, row in annualsalary.iterrows():
    if row["experience_level"] == "SE":
        annualsalary.at[index, "experience_level"] = "Senior Level/Expert"
    elif row["experience_level"] == "MI":
        annualsalary.at[index, "experience_level"] = "Medium Level/Intermediate"
    elif row["experience_level"] == "EN":
        annualsalary.at[index, "experience_level"] = "Entry Level"
    elif row["experience_level"] == "EX":
        annualsalary.at[index, "experience_level"] = "Executive Level"
```

In [63]:
```
annualsalary
```

Out[63]:

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

3755 rows × 11 columns

```
    91000  1600000   256000    72500    65720   111775    93150    21600
  4900000  1200000    21000  1799997     9272   120500    21844    22000
    76760  1672000   420000 30400000    32000   416000    40900  4450000
   423000   325000    34000    69600   435000    37000    19000    18000
    39600  1335000  1450000   190200   138350   130800   412000]
The unique values in salary_currency are: ['EUR' 'USD' 'INR' 'HKD' 'CHF' 'GBP' 'AUD' 'SGD' 'CAD' 'ILS' 'BRL' 'THB'
 'PLN' 'HUF' 'CZK' 'DKK' 'JPY' 'MXN' 'TRY' 'CLP']
The unique values in salary_in_usd are: [ 85847  30000  25500 ...  28369 412000  94665]
The unique values in employee_residence are: ['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']
The unique values in remote_ratio are: [100   0  50]
The unique values in company_location are: ['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']
The unique values in company_size are: ['L' 'S' 'M']
```

*Figure 7 Screenshot of removing the values of column experience level.*

# 3. Data Analysis

The technique which are used to make future predictions and informed data-driven by collecting, transforming, and organizing data is known as data analysis. Data analysis can be also called as the practice of working with the data to fetch the informational data, which can then used for making informed decision (Coursera Staff, 2024)

## 3.1 Question 1

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

### 3.1.1 Sum

```
In [74]: sum(annualsalary.salary_in_usd)
Out[74]: 516576814
```

*Figure 8 Screenshot of calculating the sum of column from DataFrame*

The sum of the value in 'salary_in_usd' column of the 'annualsalary' DataFrame is calculated using the function sum(). Inside the function parathesis we write the DataFrame and a column whose sum is to be generated. In the above we use 'annualsalary' DataFrame and 'salary_in_usd' column whose sum is to be generated. Then the total sum is displayed, the total salary_in_usd is 516576814.

### 3.1.2 Mean

```
In [76]: annualsalary.salary_in_usd.mean()
Out[76]: 137570.38988015978
```

*Figure 9 Screenshot of calculating the mean value.*

Mean is the average of a given number which is calculated by dividing the sum of given numbers by the total numbers. The mean value of the ‘salary_in_usd’ column is generated by using the function mean(). In above program this function is kept along side with the Dataframe and  a column name whose mean is to generated. Then the total mean is calculated and displayed, the total mean of the salary_in_usd is 137570.38988015987 (BYJU's, 2024).

### 3.1.3 Standard deviation

```
In [77]: annualsalary.salary_in_usd.std()
Out[77]: 63055.625278224084
```

*Figure 10 Screenshot of calculating the standard deviation.*

According to the data of the mean which measures the dispersion of the dataset by keeping relative to its mean is known as standard deviation. To verify whether the data point are in close proximity or spread out standard deviation is needed which is done by comparing each data point to the mean of data point. The standard deviation of the ‘salary_in_usd’ column is generated using the function std(). In above program this function is kept along side with the Dataframe and  a column name. The standard deviation is calculated and displayed, the standard deviation of the salary_in_usd is generated as 63055.625278224084 (hargrave, 2023).

### 3.1.4 Skewness

```
In [78]: annualsalary.salary_in_usd.skew()
Out[78]: 0.5364011659712974
```

*Figure 11 Screenshot of calculating the skewness.*

Skewness is the degree of the asymmetry observed in a probability distribution. Distribution can be positive and right-skewed or negative and left-skewed but if there is an normal skewness it is known as zero skewness. The standard deviation of the ‘salary_in_usd’ column is generated using the function skew(). In above program this

function is kept along side with the Dataframe and a column name. The skewness is calculated and displayed, the skewness of the salary_in_usd is generated as 0.5364011659712974 (Turney, 2022).

### 3.1.5 Kurtosis

```
In [79]: annualsalary.salary_in_usd.kurt()
Out[79]: 0.8340064594833612
```

*Figure 12 Screenshot of calculating the Kurtosis value.*

The measuring of one-tailed test distribution is known as kurtosis. keeping relative with normal distribution, the non-negative value describes the shape of the tails of distribution. The kurtosis of the 'salary_in_usd' column is generated using the function kurt(). In above program this function is kept along side with the Dataframe and a column name. The kurtosis is calculated and displayed, the kurtosis of the salary_in_usd is generated as 0.8340064594833612 (The editors of encyclopaedia britannca, 2024)

## 3.2 Question 2

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

**Correlation of all possible variables**

```
In [86]: annualsalary[['work_year','salary_in_usd','remote_ratio']].corr()
```

Out[86]:

| | 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 13 Screenshot of calculating the correlation of all possible variables.*

Correlation is a statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate). It's a common tool for describing simple relationships without making a statement about cause and effect.

The sample correlation coefficient, *r*, quantifies the strength of the relationship. Correlations are also tested for statistical significance. There are three types of correlation such as: positive linear correlation, negative linear correlation and non-linear correlation (Jmp statistical discovery, 2024).

The above figure shows the results of correlation analysis on the 'annualsalary' dataset, which contains data on employee 'work_year', 'salary_in_usd' and 'remote_ratio'. From the above table we can clearly see that each cell shows the correlation coefficient between two variables. It ranges from –1 to 1 where 1 having perfect positive correlation, -1 having perfect negative correlationa and 0 having no correlation.

In the above table, it tells us about the correlation between the variable in 'annualsalary' dataset. There is a week correlation between "workyear" and "salaryinusd" which means if employee working for a company increases then their salary also increases slightly. There is a weak negative correlation between "workyear" and "remoteratio" which means if the employee working year increases, then remote work tends to decrease slightly. There is also weak negative correlatio between "salaryinusd" and "remoteratio" which means if the employee salary increases then the remoteratio decreases. Overally we can say correlation between the variables in this dataset is weak and hence there is no strong relationship between variables.

# 4. Data Exploration

Data exploration is the first or the initial step of data analysis where data analysts use visualization data and statistical techniques to describe dataset characterizations such as size, quantity, and accuracy. They use this technique to gain the better understanding of the nature of data. In data exploration technique both manual analysis and automated data exploration techniques are included, these are the software solutions that visually explore and identify relationships between two or more data variables, structure of dataset (heavy.ai).

## 4.1 Question 1

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

**Program to find out Top 15 jobs**

```
In [77]: jobs= annualsalary['job_title'].value_counts().head(15)
         jobs

Out[77]: 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 14 Screenshot of finding top 15 jobs.*

**Bar-graph of Top 15 jobs**

```
In [98]: plt.figure(figsize=(50, 30))
         plt.xlabel("job_title",fontsize=70)
         plt.ylabel("frequency",fontsize=70)
         plt.xticks(rotation=90,fontsize=30)
         plt.yticks(fontsize=30)
         plt.bar(jobs.index, jobs.values, color='pink')

         plt.title("Distribution of top 15 jobs", fontsize=70)
         plt.show()
```

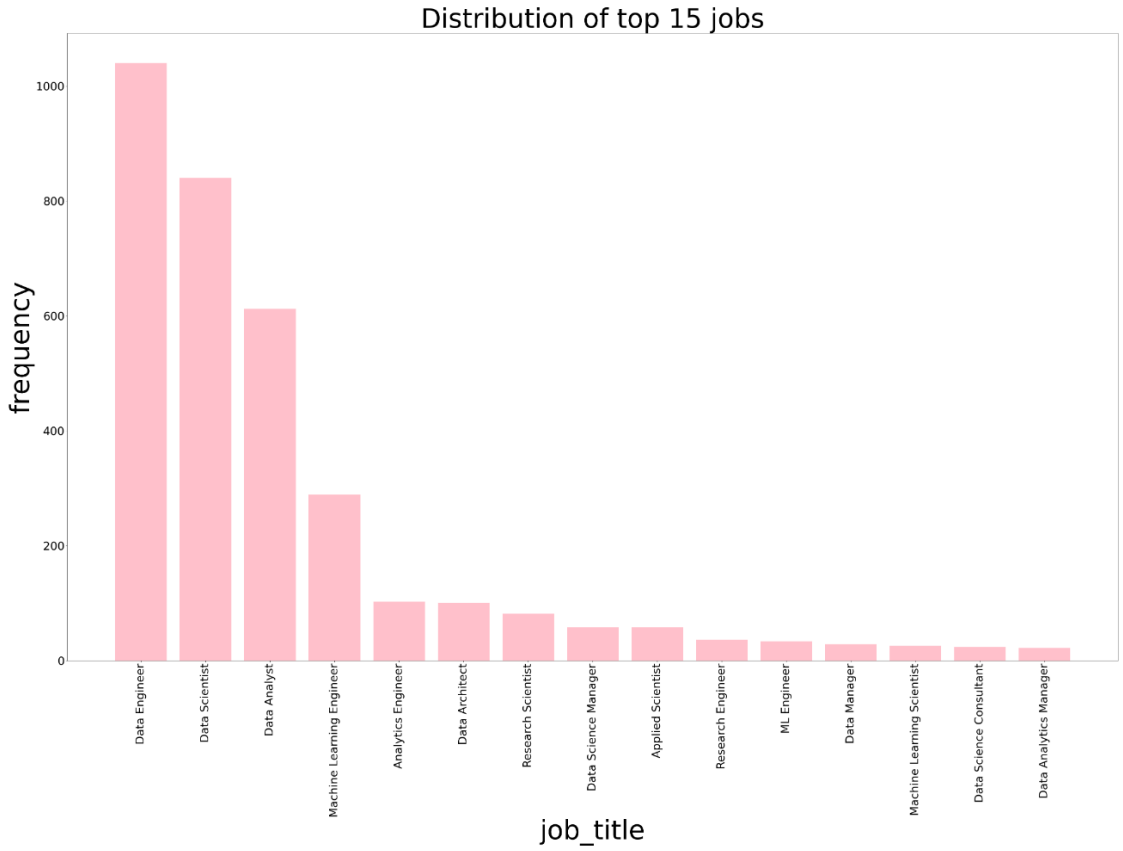


*Figure 15 Screenshot of bar graph of top 15 jobs*

The above bar graph shows the distribution of the top 15 jobs which are sorted by their frequency. Having high frequency of the job makes the job more popular. In the above bar graph the job are displayed on the x-axis whereas frequency is displayed on the y-axis. If the bar is high, the more frequent the job title appears in the data.

From the graph above, Data Engineer, Data Scientists and Data Analyst are the most demanding job titles, having frequency of 1040, 840, 612. On the other hand, job title such as Machine Learning Engineer have the low frequencies which referred as less demanding job or less common but have an important role in the field.

## 4.2 Question 2

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

**Program to find out Top 5 highest salaries**

```
In [80]: salary= annualsalary['salary_in_usd'].value_counts().head(5)
         salary
```

```
Out[80]: salary_in_usd
         100000    99
         150000    98
         120000    91
         160000    84
         130000    82
         Name: count, dtype: int64
```

*Figure 16 Screenshot of finding out top 5 highest salaries*

**Bar-graph of Top 5 highest salaries**

```
In [101]: plt.figure(figsize=(20, 10))
          plt.xlabel("salary in usd",fontsize=30)
          plt.ylabel("frequency",fontsize=30)
          plt.xticks(fontsize=15)
          plt.yticks(fontsize=15)
          plt.bar(salary.index.astype(str), salary.values,color=['orange'])
          plt.title("Salary Distribution", fontsize=40)
          plt.show()
```

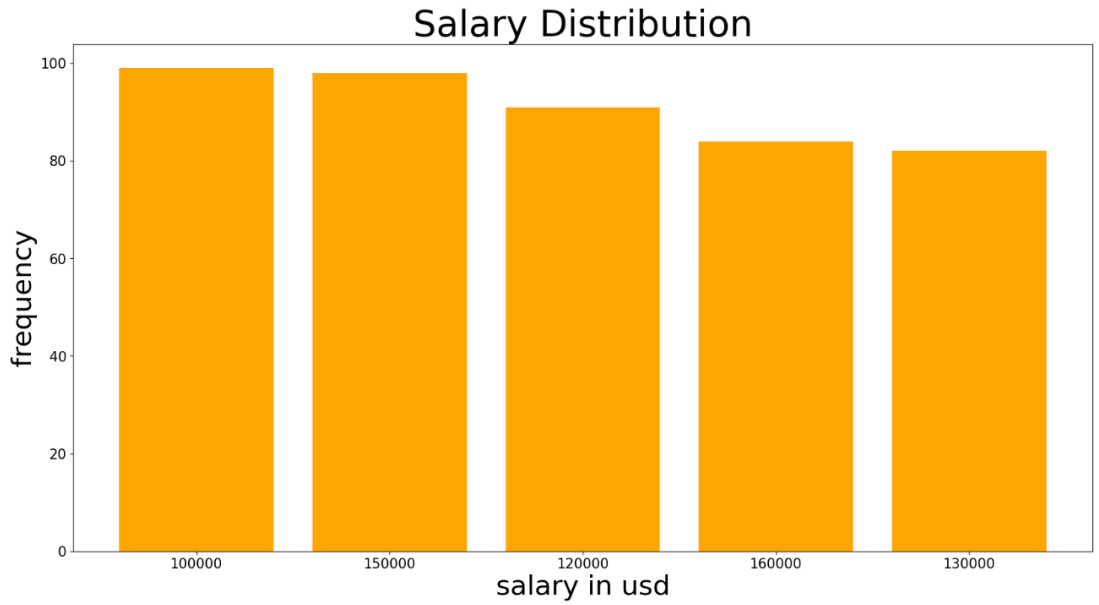


*Figure 17 Screenshot of bar graph of top 5 salaries*

From the above bar graph, we can clearly see the salary distribution of 15 jobs with respect to frequency highlighting the top 5 highest paid salaries. In the above bar graph, the x-axis of bar represents the salary in the USD having a specific range whereas y-axis of bar represents the frequency of a each salary name. The frequency can be also classified as the number of employees who earn a salary within the corresponding salary range.

## 4.3 Question 3

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

**Program to find out salaries based on experience level**

```
In [82]: experience_salary= annualsalary.groupby("experience_level")["salary_in_usd"].mean()
         experience_salary

Out[82]: experience_level
         Entry Level                  78546.284375
         Executive Level             194930.929825
         Medium Level/Intermediate   104525.939130
         Senior Level/Expert         153051.071542
         Name: salary_in_usd, dtype: float64
```

*Figure 18 Screenshot of finding out salaries based on experience level.*

**Bar-graph of salaries based on experience level**

```
In [102]: plt.figure(figsize=(20, 10))
plt.xlabel("experience_level",fontsize=30)
plt.ylabel("frequency",fontsize=30)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.bar(experience_salary.index,experience_salary.values, color=['lightgreen'])
plt.title("Salary Distribution", fontsize=40)
plt.show()
```


*Figure 19Screenshot of bar graph of salaries based on the experience level*

The above bar graph is the visual representation of the salary distribution based on the experience level. From the above bar graph, we can see that x-axis represents the different experience level which are classified as Entry Level, Executive Level, Medium Level/Intermediate and Senior Level/Expert and the y-axis represents the frequency of salary range for a given experience level.

From the above graph we can observe that the salary range is from 0 to 200,000 having low salary as 25000 and highest salary as 200000. The experience level with the highest frequency of lower salary ranges is the Entry Level, with salaries ranging from 25,000 to 75,000. The experience level with the highest frequency of higher salary ranges is the Senior Level/Expert, with salaries ranging from 125,000 to 200,000. The Medium Level/Intermediate experience has a salary range of 50,000 to 125,000, having higher frequency of salaries. The Executive Level experience has a salary range of 75,000 to 175,000, with a higher frequency of salaries.

## 4.4 Question 4

Question: 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**

```
In [105]: plt.figure(figsize=(20, 10))
plt.hist(annualsalary["salary_in_usd"], color=['purple'])
plt.xlabel("salary_in_usd",fontsize=30)
plt.ylabel("frequency",fontsize=30)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.title("Histogram of Salaries", fontsize=40)
plt.show()
```

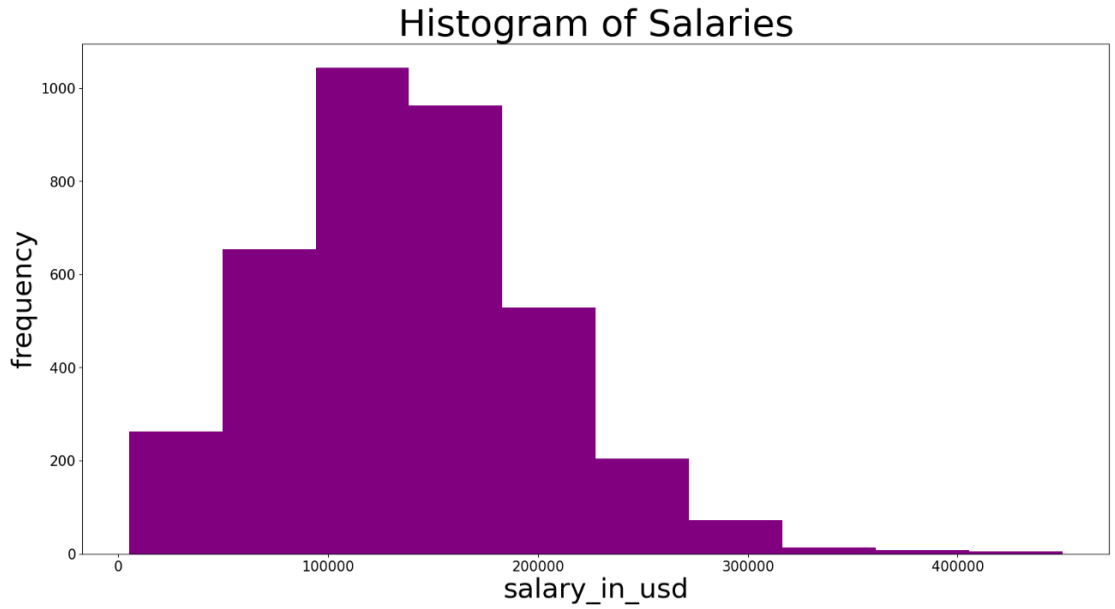


*Figure 20 Screenshot of histogram of salary*

The above histogram is the visualization of the salaries in USD of a certain group of individuals. The x-axis in the histogram represents salary_in_usd, ranging from 100,000 to 400,000 and the y-axis represents the frequency in the salary range.

From the above histogram we can clearly observe that most of the salaries lies within the 100,000 to 300,000 USD range with a significant number of individuals earning between 100,000 and 200,000 USD. There is also several individuals earning between 300,000 and 400,000. The number of individuals earning above 400,00 USD decreases rapidly, having only few individuals between 100,000 and 200,000 USD.

This histogram provides valuable insights into the salary distribution of the group, and can be used for various purposes.

**Boxplot of Salary_in_usd**

```
In [104]: plt.figure(figsize=(20, 10))
          plt.boxplot(annualsalary["salary_in_usd"])
          plt.xlabel("salary in usd",fontsize=30)
          plt.ylabel("frequency",fontsize=30)
          plt.xticks(fontsize=15)
          plt.yticks(fontsize=15)
          plt.title("Boxplot of Salaries", fontsize=40)
          plt.show()
```

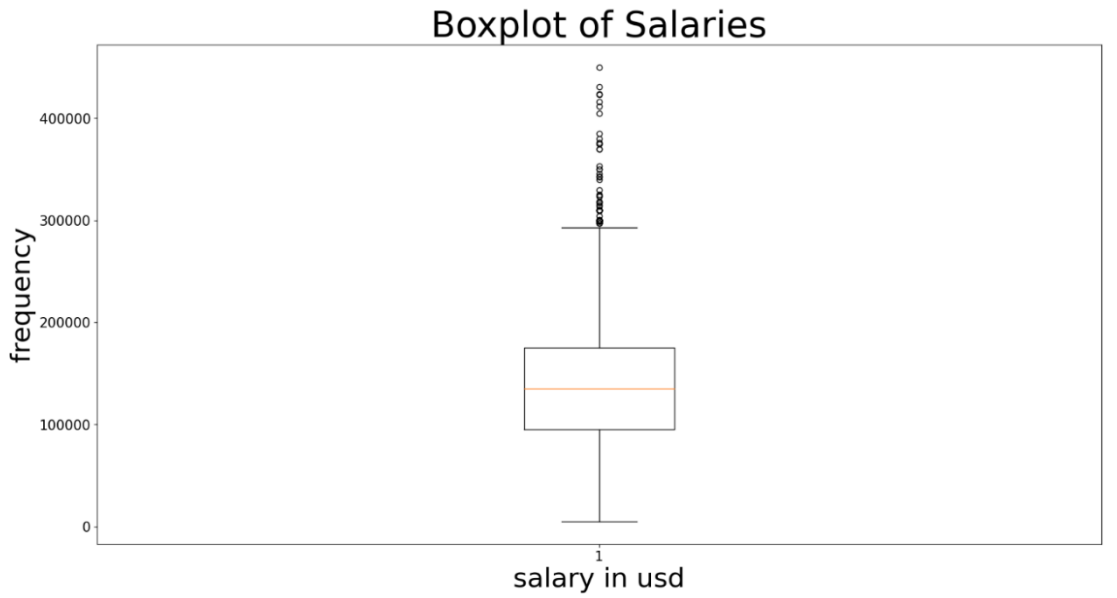


*Figure 21 Screenshot of boxplot salaries*

The above diagram shows the visual representation of the distribution of salaries in box plot. The x-axis represents the salary in USD and the y-axis represents the frequency of each salary value. The box plot displays the interquartile range (IQR) of the salaries, with the box encompassing the middle 50% of the data. The median salary is represented by the line within the box. The whiskers (line segment outside the box) extending from the box indicate the range of the data, with the upper whisker extending to the largest data point within 1.5 times the IQR above the third quartile (Q3) and the lower whisker extending to the smallest data point within 1.5 times the IQR below the first quartile (Q1).

The absence of outliers beyond the whiskers suggests that the data is relatively clean and does not contain extreme values. The box plot provides a clear and concise summary of the salary distribution.

## 5. Conclusion

In this assessment we thoroughly got to explore various libraries, functions and tools in the data set. We learned that data set can be used for various purposes such as analyzing salary trends in the data science field, comparing salaries across different jobs experience levels, locations, exploring the impact of remote work.

## 6. References

- BYJU's, 2024. *BYJU's.* [Online]
  Available at: https://byjus.com/maths/mean/#:~:text=Definition%20of%20Mean%20in%20Statistics,observations%2FTotal%20number%20of%20observations)
  [Accessed Saturday May 2024].
- Coursera Staff, 2024. *Coursera.* [Online]
  Available at: https://www.coursera.org/articles/what-is-data-analysis-with-examples
  [Accessed Saturday May 2024].
- geeksforgeeks, 2023. *geeksforgeeks.* [Online]
  Available at: https://www.geeksforgeeks.org/what-is-dataset/
  [Accessed 01 May 2024].
- hargrave, M., 2023. *investopedia.* [Online]
  Available at: https://www.investopedia.com/terms/s/standarddeviation.asp#:~:text=What%20Does%20Standard%20Deviation%20Tell,whether%20they%20are%20spread%20out.
  [Accessed Saturday May 2024].
- Jmp statistical discovery, 2024. *jmp.* [Online]
  Available at: https://www.jmp.com/en_ca/statistics-knowledge-portal/what-is-correlation.html#:~:text=What%20is%20correlation%3F,statement%20about%20cause%20and%20effect.
  [Accessed Saturday May 2024].
- The editors of encyclopaedia britannca, 2024. *Britannica.* [Online]
  Available at: https://www.britannica.com/topic/kurtosis-statistics
  [Accessed Saturday May 2024].
- Turney, S., 2022. *scribbr.* [Online]
  Available at: https://www.scribbr.com/statistics/skewness/
  [Accessed Saturday May 2024].