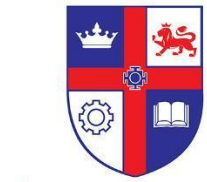
LONDON METROPOLITAN UNIVERSITY



Islington college
(इस्लिङ्टन कलेज)

CC5067NI-Smart Data Discovery

60% Individual Coursework

2023-24 Spring

Student Name: Sudam Pant

London Met ID: 22067871

College ID: np01cp4a220252@islingtoncollege.edu.np

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Monday, May 13, 2024

Word Count: 2800

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 Tables

Table Of Figures

Introduction

The job of a data scientist has become very popular in recent years, as more and more companies need help making sense of their data. This report is about looking at how much money does scientists make in the United States. We're going to use a range of data-on-data scientist salaries to find out what factors influence how much they get paid.

Our analysis involves a wide range of research, from analyzing the distribution of salaries across different experience levels, educational backgrounds, and geographical locations. We also explore the complex correlations and relationships between salaries and various factors such as job titles, skills and industry sectors. Using regression techniques, we delve deeper to understand the subtle influence of specific variables on data scientist salaries. In time series analysis, we aim to identify temporal trends and shifts in compensation levels over time, providing a holistic view of the evolving landscape of data scientist salaries.

We aim to create insightful visual representations that effectively communicate our findings and facilitate clear interpretation, with a strong emphasis on data visualization. In the end, this report is about helping those interested in working in data science to make smart decisions about their future and their expected earnings. So, let's take a dive into the report and explore what factors influence data scientist salaries in the US.

Data Understanding

The first step in understanding the data is to read and understand the dataset on computing salaries. We identify the variables, the types of data and the overall structure of the data set.

The dataset provided contains information on various factors that influence Data Scientist salaries, such as experience, work level, job title and many more. The aim of this analysis is to gain a better understanding of the elements that influence Data Scientist salaries and to discover any patterns or trends within the data.

This dataset will be used to prepare for further data mining and analysis.

Here's a table that summarizes the relevant information for each of the fields in the data set:

S.No.	Column Name	Description	Data Type
1	Work Year	Year in which the data was recorded	Categorical
2	Experience Level	Seniority level of the data scientist (e.g., Entry-Level, Mid-Level, Senior Level)	Categorical
3	Employment Type	Type of employment (e.g., Full-Time, Part-Time, Contract)	Categorical
4	Job Title	Specific role of the data scientist (e.g., Data Analyst, Data Scientist, Machine Learning Engineer)	Categorical
5	Salary	Annual salary of the data scientist (USD)	Numerical
6	Salary Currency	Currency in which the salary is originally reported	Categorical
7	Salary In USD	Converted salary in USD for consistency	Numerical
8	Employee Residence	Geographic location of the data scientist's residence	Categorical
9	Remote Ratio	Percentage of time the data scientist works remotely	Numerical
10	Company Location	Geographic location of the data scientist's employer	Categorical
11	Company Size	Size of the company employing the data scientist (e.g., Small, Medium, Large)	Categorical

Table 1: Table of every column of dataset

Data Preparation

This phase is about transforming the raw data into a format which is suitable for further analysis. We start by loading the data into a pandas Data Frame, a fundamental structure for data manipulation in Python. Then, we remove irrelevant columns, such as salary and salary currency, to streamline the analysis. Additionally, we address missing values by removing rows containing Nan entries to ensure data integrity. Finally, we identify and handle potential duplicates by checking for duplicate rows within the Data Frame.

Also, we explore the unique values present in each column, providing insights into the data's diversity and distribution. Finally, we rename the experience level columns to enhance readability and clarity, using more descriptive labels like "Senior Level/Expert" and "Entry Level."

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

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import seaborn as sns
```

```
[2]:
#importing given salary csv file in the form of dataframe
```

```
[3]:
df = pd.read_csv('Salary.csv')
```

Figure 1: First Step Of pyhton programming

In the figure above, we first import the libraries that will be needed for the tasks below. Importing the Pandas library for data manipulation and Matplotlib for data visualization.

To load data into a Pandas Data Frame in Python, we can use the Pandas library read_csv() function to read data from a CSV file or other formats like Excel, JSON, SQL databases, etc. The DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. It is similar to a spreadsheet or SQL table, where each column can be of a different data type (integer, string, float, etc.). (pandas, n.d.)

So, this is our data in data frame.

```
df
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	compan
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: Data Frame

To confirm the presence of null values, `df.isnull()` is used to check for their existence, while `df.shape` returns the dimensions of the Data Frame, revealing the number of rows and columns it contains.

```
df.shape
```

```
(3755, 11)
```

```
df.isnull()
```

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

3755 rows × 11 columns

Figure 3 : Finding null values

```
[6]: df.isnull().sum()
```

```
[6]: 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
     dtype: int64
```

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

```
[7]:
df = df.drop(['salary', 'salary_currency'], axis=1)

[40]:
df.shape

[40]:
(3755, 9)
```

Figure 4: Dropping columns

We Drop the "salary" and "salary_currency" columns as they are not relevant for the analysis.

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

To check for the presence of NaN values in the above Data Frame, we use `df. isna()`. As no NaN values are detected, there's no need to remove them. However, if NaN values were detected, they could be removed using the `data. dropna()` method.

```
[8]: df.isna()
```

[8]:

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

Figure 5: Checking NA values

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

Checking if there are duplicate values or not. By using df.duplicated.()

```
# Program To Check Duplicates in the database
```

```
[10]:
df.duplicated()
```

```
[10]:
0       False
1       False
2       False
3       False
4       False
        ...
3750    False
3751    False
3752    False
3753    False
3754    False
Length: 3755, dtype: bool

[11]:
```

Figure 6: Check Duplicates

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

Loop through each column and print the unique values present in that column to understand the data diversity.

```
# Program To See the unique values from all the columns
```

[12]:

```
unique_values = {}
for column in df.columns:
    unique_values[column] = df[column].unique()
```

```
print("\nUnique values from each column:")
for column, values in unique_values.items():
    print(column + ":", values)
```

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

Figure 7: Printing the unique values.

6. Rename the experience level columns as below.

SE – Senior Level/Expert

MI – Medium Level/Intermediate

EN – Entry Level

EX – Executive Level

We use a level mapping dictionary to rename the experience levels according to the given question. This mapping facilitates the grouping of the data appropriately. Once the mapping is in place, we replace the experience level data in the Data Frame with the mapped values. This process ensures that the experience levels are accurately represented in the data frame in accordance with the specified mappings.

```
#Renaming the experience level columns as below.
#SE - Senior Level/Expert
#MI - Medium Level/Intermediate
#EN - Entry Level
#EX - Executive Level
```

[15]:

```
level_mapping = {
    'SE': 'Senior Level/Expert',
    'MI': ' Medium Level/Intermediate',
    'EN': 'Entry Level',
    'EX':' Executive Level'
}
```

[16]:

```
df['experience_level'] = df['experience_level'].replace(level_mapping)

print("\nDF after renaming 'experience_level' column:")
print(df)
```

Figure 8: Renaming The experience level column.

```
DF after renaming 'experience_level' column:
      work_year            experience_level employment_type  \
0          2023         Senior Level/Expert              FT
1          2023   Medium Level/Intermediate              CT
2          2023   Medium Level/Intermediate              CT
3          2023         Senior Level/Expert              FT
4          2023         Senior Level/Expert              FT
...         ...                         ...             ...
3750       2020         Senior Level/Expert              FT
3751       2021   Medium Level/Intermediate              FT
3752       2020                Entry Level              FT
3753       2020                Entry Level              CT
3754       2021         Senior Level/Expert              FT
```

Figure 9: Output After renaming experience level column.

Data Analysis

Once the data has been prepared, we move on to quantitative analysis. The first step is to calculate summary statistics for a chosen variable, such as the sum, mean, standard deviation, skewness, and kurtosis. These measures provide a concise overview of the variable's central tendency, dispersion, and distribution shape.

The second step is to calculate the correlation between all variables, revealing the strength and direction of linear relationships between them. This allows us to identify the key variables that are strongly correlated with salary, providing us with potential explanatory factors.

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

We can use the describe() function together with additional functions from the scipy.stats module to compute summary statistics (sum, mean, standard deviation, skewness and kurtosis) for a chosen variable. Here's how we can do this, where we replace "chosen_variable" with the actual name of the variable.

```
#Data Analysis Part
# Summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.
```

```
from scipy.stats import skew, kurtosis
```

```
# Choosing a variable for summary statistics
chosen_variable = 'salary_in_usd'

# Compute summary statistics
summary_stats = df[chosen_variable].describe()
variable_sum = df[chosen_variable].sum()
variable_skewness = skew(df[chosen_variable])
variable_kurtosis = kurtosis(df[chosen_variable])

# Print summary statistics
print("Summary statistics for variable", chosen_variable)
print(summary_stats)
print("\nSum:", variable_sum)
print("Mean:", summary_stats['mean'])
print("Standard deviation:", summary_stats['std'])
print("Skewness:", variable_skewness)
print("Kurtosis:", variable_kurtosis)
```

Figure 10: Statistics.

```
Summary statistics for variable salary_in_usd
count       3755.000000
mean     137570.389880
std       63055.625278
min        5132.000000
25%       95000.000000
50%      135000.000000
75%      175000.000000
max      450000.000000
Name: salary_in_usd, dtype: float64

Sum: 516576814
Mean: 137570.38988015978
Standard deviation: 63055.625278224084
Skewness: 0.5361868674235593
Kurtosis: 0.8312989014514311
```

Figure 11: Output of statistics data

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

For the calculation of correlation coefficients between all pairs of variables in the data frame, we can use the corr() function in Pandas.

[20]:

```
# program to calculate and show correlation of all variables.
```

[21]:

```
df_numeric = df.apply(lambda x: pd.factorize(x)[0] if x.dtype == 'object' else x)
correlation_matrix = df_numeric.corr()

print("Correlation matrix of all variables:")
print(correlation_matrix)
```

Figure 12: Calculation Of Correlation

```
Correlation matrix of all variables:
                    work_year  experience_level  employment_type  job_title  \
work_year            1.000000         -0.167648        -0.115248  -0.156954
experience_level    -0.167648          1.000000         0.125208   0.127764
employment_type     -0.115248          0.125208         1.000000   0.060485
job_title           -0.156954          0.127764         0.060485   1.000000
salary_in_usd        0.228290         -0.243669        -0.126923  -0.067952
employee_residence  -0.279475          0.208346         0.233934   0.187946
remote_ratio        -0.236430          0.044598         0.059242   0.064941
company_location    -0.261057          0.189563         0.129597   0.175600
company_size         0.389401         -0.210922        -0.088431  -0.183533

                    salary_in_usd  employee_residence  remote_ratio  \
work_year                0.228290           -0.279475     -0.236430
experience_level        -0.243669            0.208346      0.044598
employment_type         -0.126923            0.233934      0.059242
job_title               -0.067952            0.187946      0.064941
salary_in_usd            1.000000           -0.302527     -0.064171
employee_residence      -0.302527            1.000000      0.100911
remote_ratio            -0.064171            0.100911      1.000000
company_location        -0.300085            0.822244      0.074603
company_size             0.166600           -0.239687     -0.138520

                    company_location  company_size
work_year                  -0.261057      0.389401
experience_level            0.189563     -0.210922
employment_type             0.129597     -0.088431
job_title                   0.175600     -0.183533
salary_in_usd              -0.300085      0.166600
employee_residence          0.822244     -0.239687
remote_ratio                0.074603     -0.138520
company_location            1.000000     -0.232940
company_size               -0.232940      1.000000
```

Figure 13: Output of correlation of all variables

Data Exploration

The next step is to explore the data visually. This is the best way to uncover patterns and insights. We start by identifying the fifteen most common jobs in the dataset. Next, we create a bar chart to show where data scientists are working. This makes it easy to see the job landscape.

We can identify the job with the highest average salary and present this information in a bar chart. This allows us to identify the best jobs in data science.

Then we look at the relationship between experience level and salary. We calculate the average salary for each experience level and present this information in a bar chart. This visualisation clearly shows how salaries vary with increasing experience.

After that we select a particular variable and create a histogram and box plot to gain a deeper insight into its distribution. The histogram provides a comprehensive frequency distribution, while the box plot shows the variable's median, quartiles and outliers. This provides a definitive understanding of the characteristics of the variable.

In Our Project Data Visualization Include:

a. **Histogram:**

A histogram is a graphical representation that shows the distribution of values of a numerical variable by displaying them as a series of adjacent bars. Each bar spans a range of values known as a bin or class, with the height of the bar indicating the frequency of data points falling within that range. For example, in the histogram shown, which shows response times for support tickets, each bar represents an hour and shows the number of tickets within that time period. Notably, the histogram shows distinct peaks that highlight the prevalence of response times within certain hour ranges, providing insight into the distribution beyond mere statistical summaries such as mean and standard deviation. (YI, n.d.).

b. Bar Graphs:

A bar graph is a graphical representation of data using rectangular bars whose lengths indicate the size of the corresponding data points. These bars can be arranged vertically or horizontally, providing flexibility in visual presentation. Commonly referred to as column charts or bar graphs, they provide an easy way to compare different categories or illustrate changes over time. By encoding data in the length of the bars, bar graphs allow relative magnitudes and trends to be quickly understood. This makes them valuable tools for presenting information concisely and facilitating the effective communication of insights across different domains and disciplines. (Geeks, 2024)

c. Box Plots:

In descriptive statistics, a scatterplot, also known as a box and whisker plot, serves as a valuable tool for analyzing explanatory data. This type of graph effectively illustrates the distribution of numerical data and skewness through the visual representation of key summary statistics. The box plot displays the five-digit summary of the data set, which includes the minimum value, the first quartile (lower), the median, the third quartile (upper) and the maximum value. By presenting these key metrics, the box plot provides an insight into the central tendency, spread and variability of the data, and helps to identify outliers and patterns within the data set. (Mcleod, 2023).

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

First Identify the top 15 job titles and create a bar chart showing the frequency distribution of data scientists across these job titles.

```
# Now lets explore our data
```

We need to import matplotlib.pyplot library to create bar graphs and other visual representations

```
[26]:
import matplotlib.pyplot as plt
```

```
# top 15 jobs

top_jobs = df['job_title'].value_counts().head(15)
```

```
[30]:
# Plotting the bar graph
plt.figure(figsize=(10, 6))
top_jobs.plot(kind='bar', color='navy', edgecolor='black', linewidth=2)
plt.xlabel('Bar Diagram of Top 15 Jobs')
plt.ylabel('Sales')
plt.title('Sales based on Top Jobs')
plt.xticks(rotation=85)
plt.show()
```

Figure 14: Top 15 jobs

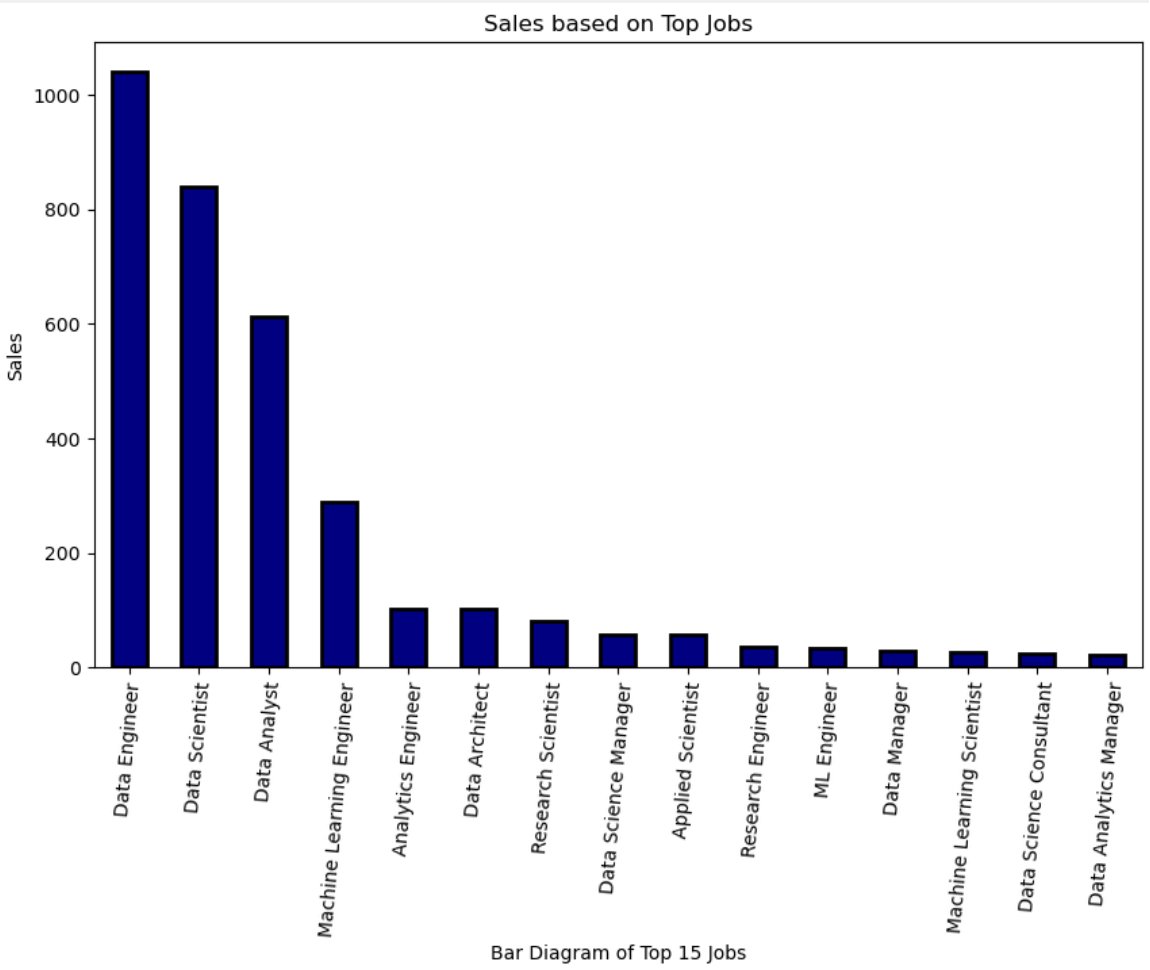


Figure 15: Bar Diagram Of Top 15 jobs

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

First Identify the job with the highest average salary and visualise this information using a bar chart. Select two columns: Job Title and Salary in usd. First we need to find the average salary per job based on salary in usd. Then we select the top 15 jobs.

```
# Finding Top paid jobs on the basis of salaries in usd

average_salary_per_job = df.groupby('job_title')['salary_in_usd'].mean()
```

[25]:

```
top_jobs = average_salary_per_job.nlargest(15)
top_jobs
```

Figure 16: Top paid Jobs

```
job_title
Data Science Tech Lead                 375000.000000
Cloud Data Architect                   250000.000000
Data Lead                              212500.000000
Data Analytics Lead                    211254.500000
Principal Data Scientist               198171.125000
Director of Data Science               195140.727273
Principal Data Engineer                192500.000000
Machine Learning Software Engineer     192420.000000
Data Science Manager                   191278.775862
Applied Scientist                      190264.482759
Principal Machine Learning Engineer    190000.000000
Head of Data                           183857.500000
Data Infrastructure Engineer           175051.666667
Business Intelligence Engineer         174150.000000
Machine Learning Scientist             163220.076923
Name: salary_in_usd, dtype: float64
```

Figure 17: Results Of top Paid Jobs

```
[27]:
#plotting data in bar diagram

[28]:
plt.figure(figsize=(12, 8))
top_jobs.plot(kind='bar', color='Maroon', edgecolor='black', linewidth=2.5)
plt.xlabel('Top 15 Jobs by Salary (in USD)',color='Maroon')
plt.ylabel('Average Salary (in USD)', color='Maroon')
plt.title('Top 15 Jobs based on Salaries (in USD)', color='Green')
plt.xticks(rotation=80, color='darkolivegreen' )
plt.show()
```

Figure 18: Code for bar diagram

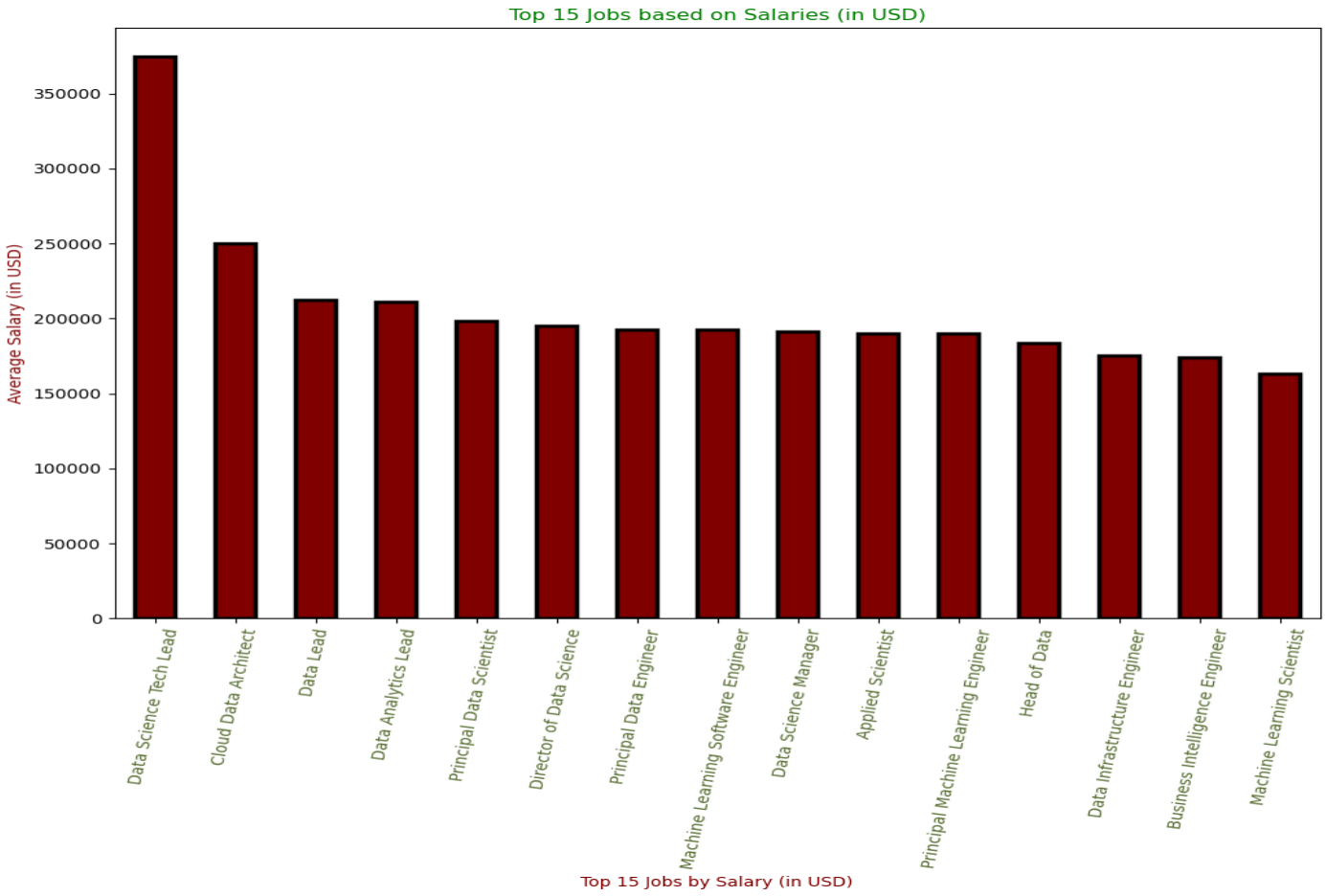


Figure 19: Bar Diagram Of top Paid jobs

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

Calculate the average salaries for each experience level and present this information in a bar graph.

```
[31]: # Calculate average salary for each experience level
average_salary_per_experience = df.groupby('experience_level')['salary_in_usd'].mean()
```

```
[32]: # Plotting the bar graph with background colors
plt.figure(figsize=(10, 6))
average_salary_per_experience.plot(kind='bar', color='Maroon', edgecolor='black', linewidth=3, facecolor='peachpuff')
plt.xlabel('Experience Level')
plt.ylabel('Average Salary (in USD)')
plt.title('Average Salary Based on Experience Level')
plt.xticks(rotation=90)
plt.show()
```

Figure 20: Average Salaries

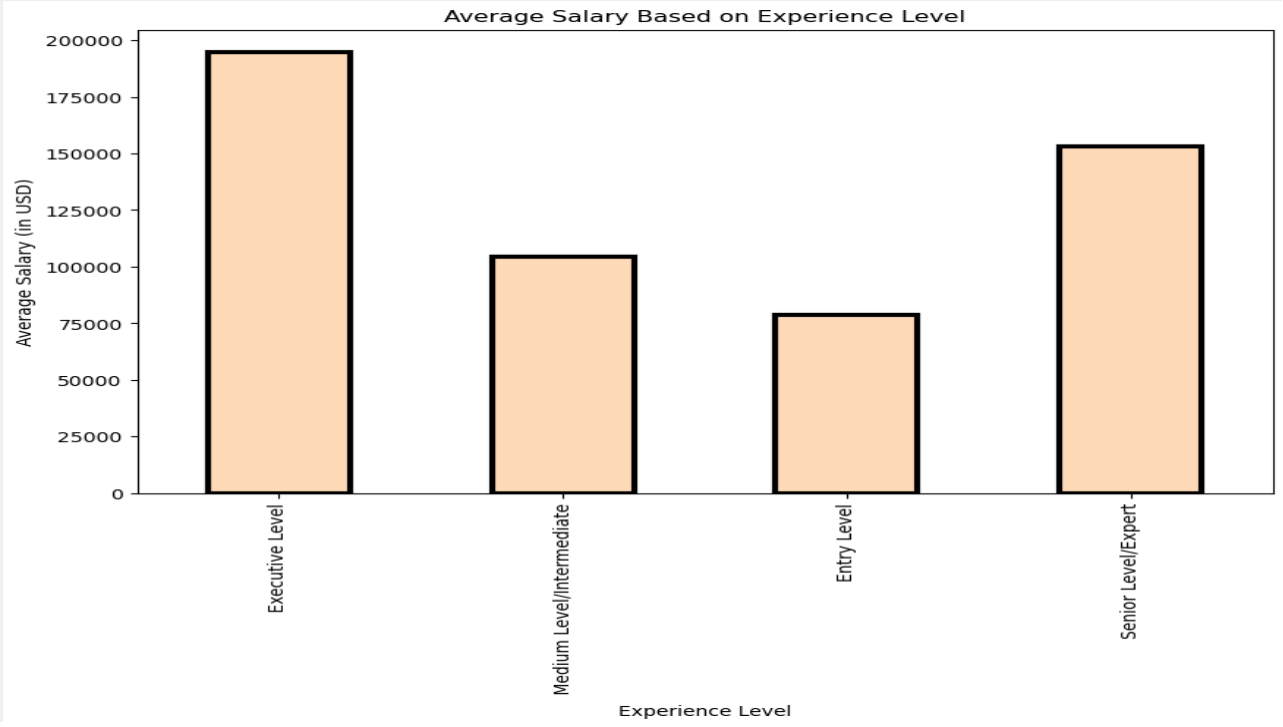


Figure 21: Average Salary Based On experience level

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

First, we write some code to find out the data types. Then we find out the columns containing int and float values. It is done to show how to find columns with numeric values for histogram in case of large data.

```
#Choosing the numerical column for histogram and Box Plots
df.dtypes
```

```
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary_in_usd          int64
employee_residence    object
remote_ratio           int64
company_location      object
company_size          object
dtype: object
```

```
numerical_columns = df.select_dtypes(include=['int', 'float']).columns
print("Numerical columns:")
print(numerical_columns)
```

```
Numerical columns:
Index(['work_year', 'salary_in_usd', 'remote_ratio'], dtype='object')
```

Figure 22: Finding Numerical columns

i. Histogram For Work Year

In this figure we have used dropna to drop all null or missing values in the Work Year and Salary in USD columns. After that we have set the figure size, colour, caption and finally the title. Then we can see the histogram of the working year.

[35]:
```
# Filter out missing values
df.dropna(subset=['work_year', 'salary_in_usd'])

# Plotting histogram for work_year
plt.figure(figsize=(12, 6))
plt.hist(df['work_year'], color='olive', edgecolor='black', bins=20)
plt.xlabel('Years of Work Experience')
plt.ylabel('Frequency')
plt.title('Histogram of Work Experience')
plt.show()
```

Figure 23: Setting up for histogram

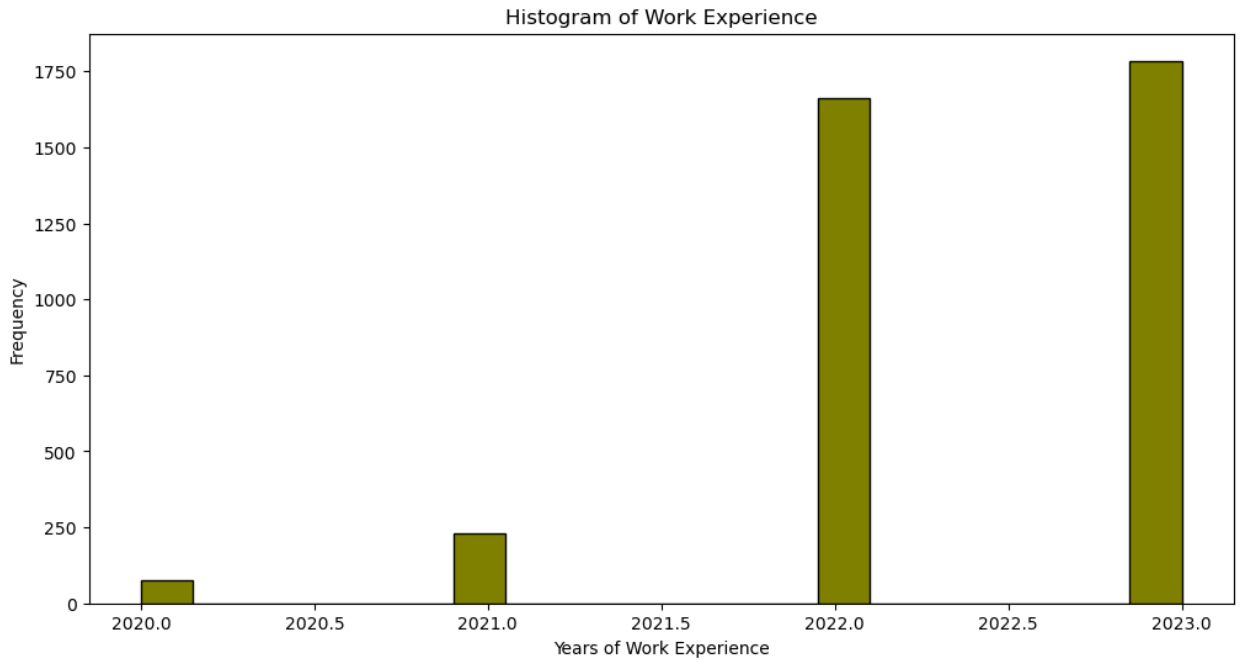


Figure 24: Histogram of work year

ii. Histogram For Salary in USD

```
# Plotting histogram for salary_in_usd

plt.figure(figsize=(12, 6))
plt.hist(df['salary_in_usd'], color='thistle', edgecolor='black', bins=20)
plt.xlabel('Salary in USD')
plt.ylabel('Frequency')
plt.title('Histogram of Salary in USD')


plt.hist
plt.show()
```

Figure 25: Code for histogram of salary in usd

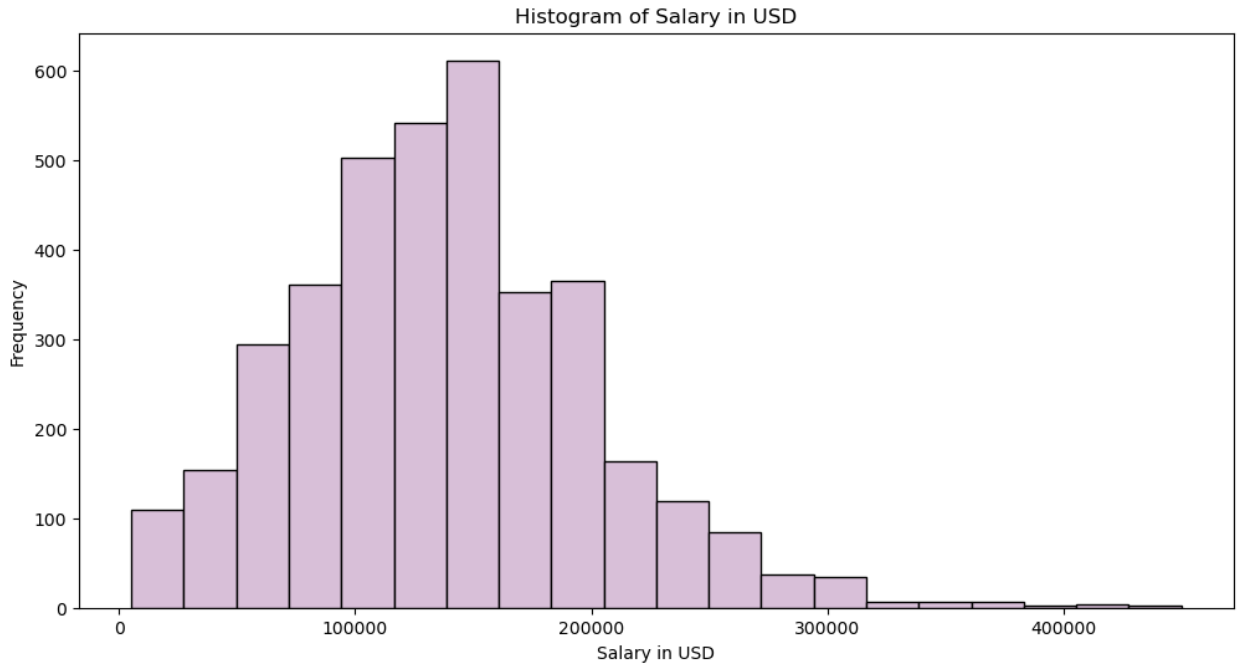


Figure 26: Histogram Of salary in usd

iii. Box Plotting of work year.

```
# Box plotting after filtering out the missing values
df.dropna(subset=['work_year'])
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 2)
sns.boxplot(x=df['work_year'], orient='h', palette='BuGn_r', linewidth=1.5)
plt.xlabel('year')
plt.ylabel('Frequency')
plt.title('Box Plot of Work Year')
plt.show()
```

Figure 27: Code for box plotting of work year

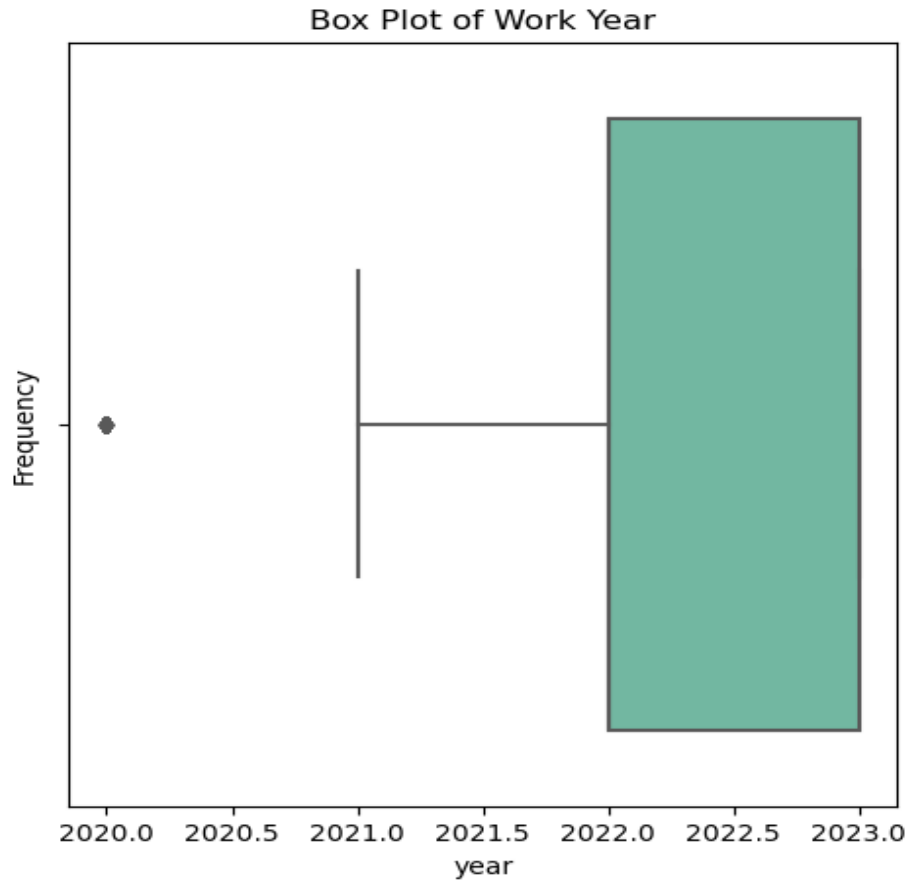


Figure 28: Box Plotting of people by work_year

iv. Box Plot Of Salary in USD

```
43]: import matplotlib.pyplot as plt
     import seaborn as sns

     plt.figure(figsize=(12, 6))
     plt.subplot(1, 2, 2)
     sns.boxplot(data=df['salary_in_usd'], orient='h', palette='magma', linewidth=1.5, )
     plt.xlabel('Salary in USD')
     plt.ylabel('values')
     plt.title('Box Plot of Salary in USD')
     plt.show()
```

Figure 29:Code for box plot of salary in USD

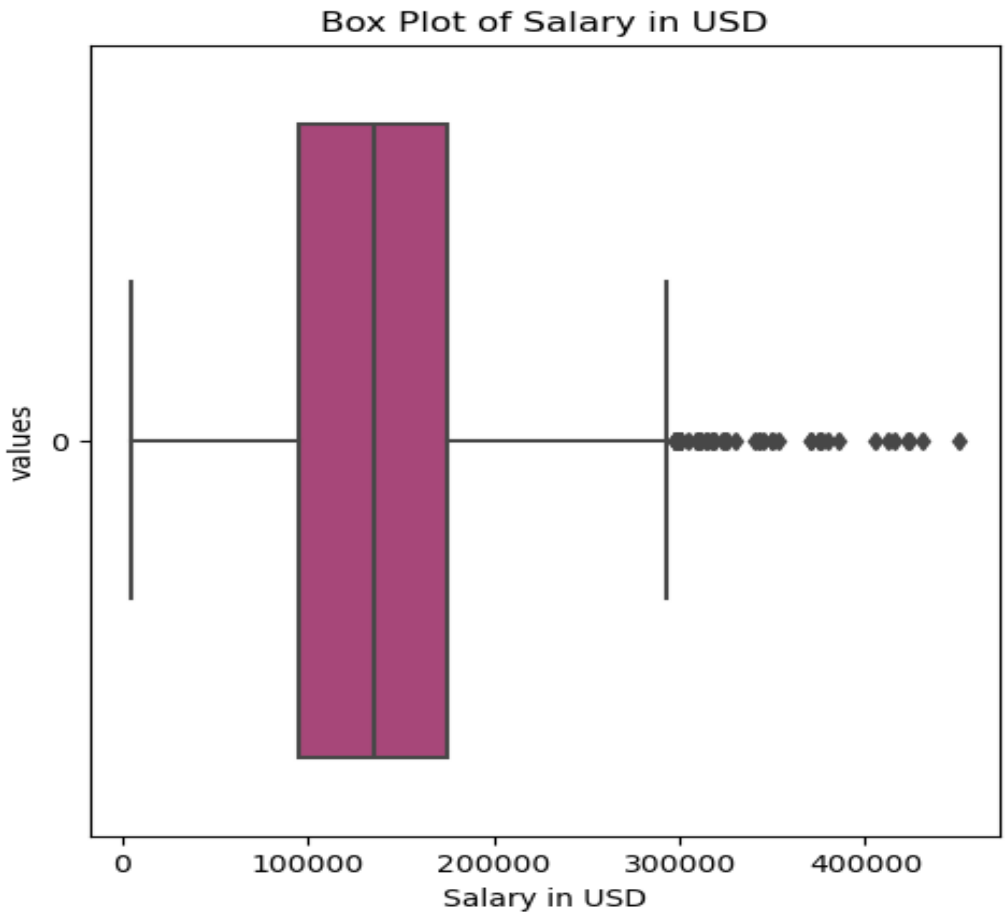


Figure 30: Box Plot Of Salary In USD

Conclusion

Our analysis of data scientist salaries in the US has provided valuable insights into the complex factors that influence compensation in this rapidly growing field. Through a comprehensive examination of salary distributions, relationships between variables, and trends over time, we've gained a deeper understanding of what drives data scientist salaries. Individuals can now make more informed decisions about their career paths and negotiate better salaries armed with this knowledge.

Our key findings highlight the significant impact of experience, education, skills and location on how much Data Scientists get paid. It's clear that those with more experience tend to command higher salaries, while those with advanced degrees tend to pay more. Likewise, those who have advanced qualifications, such as MSc or PhD, tend to earn more than those who have only an undergraduate qualification. In addition, earning potential can be significantly boosted by possessing specific skills, such as machine learning, artificial intelligence and big data analytics. Geographical location also plays a key role: Salaries tend to be higher in tech hubs and major cities.

In conclusion, our research provides valuable insights into data scientist salaries. It serves as a comprehensive resource for individuals interested in the field. By sharing our findings, our aim is to empower aspiring data scientists to make informed decisions about their career paths and advocate for fair compensation. As the field of data science continues to evolve, staying informed and adaptable will be key to long-term success in this dynamic and rewarding profession.

Bibliography

Geeks, G. F., 2024. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/bar-graph/
[Accessed 2024].

Mcleod, S., 2023. *SimplyPsychology.* [Online]
Available at: https://www.simplypsychology.org/boxplots.html
[Accessed 2024].

pandas, n.d. *pandas.* [Online]
Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
[Accessed 2024].

YI, M., n.d. *atlassian.* [Online]
Available at: https://www.atlassian.com/data/charts/histogram-complete-guide
[Accessed 2024].