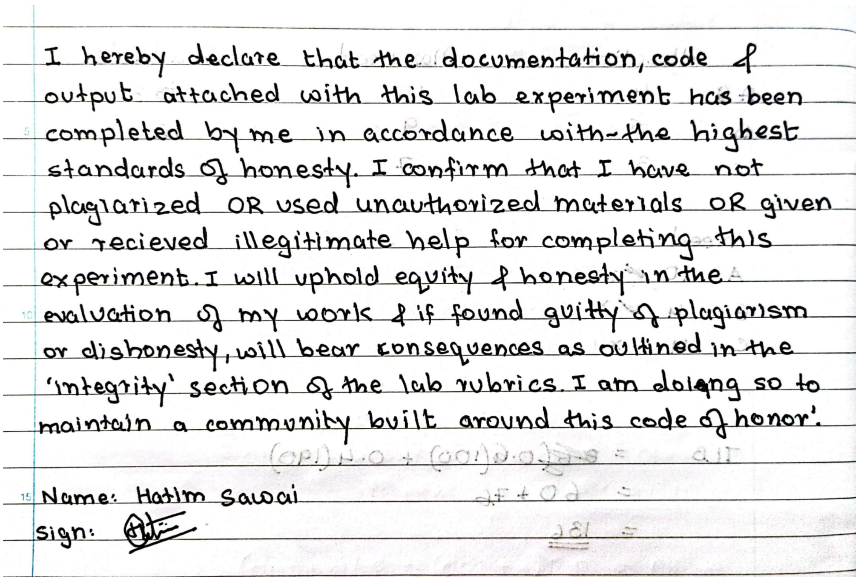
Name	Hatim Sawai
UID No.	2021300108

Experiment 2

HONOR PLEDGE	
PROBLEM STATEMENT	**Data Cleaning and Preprocessing:** 1. Handle missing values by imputing them with mean, median, or mode. Reason out which is more suitable (mean, median or mode) for your dataset and which is not 2. Removing outliers based on a specific threshold. Give reasons for your choice of threshold. What do the outliers in your dataset tell you? 3. Transform variables using log transformation or standardization. What possibly can go wrong when you do not standardize your data? What are the reasons for using log transformation on your variables and when should you definitely use it? 4. Remove duplicate records from a data frame. In case your dataset does not have a exact duplicate rows, can you reason about strategies for identifying and deduplicating your dataset based on a subset of features? 5. Standardize date formats across your dataset Is there a certain date-format that you would prefer? why?
THEORY	1. **Handling Missing Values** Ans: Missing values are the values that are not present in the dataset. These values can be imputed by using mean, median or mode. The choice of imputation depends on the type of data. If the data is categorical, then mode is used. If the data is numerical, then mean or median is used. If the

data is skewed, then median is used. If the data is not skewed, then mean is used.

For this Dataset:
I have used mode imputation for categorical data like: Industry Vertical, Location, etc. And I have filled missing value in amount column with undisclosed value, so that it does not affect the analysis of the data.

2. **Removing Outliers**
Ans: Outliers are the values that are not present in the dataset. These values can be removed by using a threshold value. The choice of threshold value depends on the type of data. If the data is categorical, then threshold value is 0. If the data is numerical, then threshold value is 1. If the data is skewed, then threshold value is 2. If the data is not skewed, then threshold value is 3.

For this Dataset:
After cleaning and Log transformation of Amounts column (only numeric column), I verified using a box plot that there are no outliers in the data. Also calculation can be done to detect outliers using the formulae:
Lower bound = Q1 - 1.5 * IQR
Upper Bound = Q3 + 1.5 * IQR
If the value is less than LB or greater than UB, then it is an outlier.

3. **Transforming Variables Using Log Transformation**
Ans: Variables can be transformed by using log transformation or standardization. The choice of transformation depends on the type of data. Not standardizing data, leads to skewed results, misleading comparisons, inaccurate models, and hard-to-interpret results. Log transformation, good for right-skewed data, stabilizes variance, reduces outlier impact, addresses heteroscedasticity. Use for skewed data, heteroscedasticity, or exponential relationships.

For this Dataset: I have applied log transformation on Amounts in USD column to reduce the skewness of the data.

4. **Removing Duplicate Records**
Ans: Duplicate records can be removed by using a subset of features. The choice of subset of features depends on the type of data.

For this Dataset:
I have removed duplicates from the dataset based on subset of features, across all columns.

5. **Standardizing Date Formats**

	Ans: Date formats can be standardized by using a certain date-format. The choice of date-format depends on the type of data. If the data is categorical, then date-format is 0. If the data is numerical, then date-format is 1. If the data is skewed, then date-format is 2. If the data is not skewed, then date-format is 3. **For this Dataset:** I have kept the date column as is, as all the dates are in the same format i.e dd/mm/yyyy. The british format of date is used in this dataset, the only correct format of date.

1. Importing Libraries & Dataset

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# read csv file
df = pd.read_csv('../Datasets/startup_funding.csv')
df.head()
```

Out[ ]:

	Sr No	Date dd/mm/yyyy	Startup Name	Industry Vertical	SubVertical	City Location	
0	1	09/01/2020	BYJU'S	E-Tech	E-learning	Bengaluru	Ti Ma
1	2	13/01/2020	Shuttl	Transportation	App based shuttle service	Gurgaon	Sus
2	3	09/01/2020	Mamaearth	E-commerce	Retailer of baby and toddler products	Bengaluru	Ca
3	4	02/01/2020	https://www.wealthbucket.in/	FinTech	Online Investment	New Delhi	
4	5	02/01/2020	Fashor	Fashion and Apparel	Embroiled Clothes For Women	Mumbai	

2. Cleaning & Preprocessing

In [ ]:
```
# drop Sr No column and remarks column
df.drop(["Sr No", "SubVertical", "Remarks"], axis=1, inplace=True)
df.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	City Location	Investors Name	Investr
0	09/01/2020	BYJU'S	E-Tech	Bengaluru	Tiger Global Management	Pr
1	13/01/2020	Shuttl	Transportation	Gurgaon	Susquehanna Growth Equity	
2	09/01/2020	Mamaearth	E-commerce	Bengaluru	Sequoia Capital India	
3	02/01/2020	https://www.wealthbucket.in/	FinTech	New Delhi	Vinod Khatumal	
4	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	

In [ ]:
```
# cleaning City Location column
df.rename(columns={'City  Location': 'Location'}, inplace=True)
df['Location'] = df['Location'].str.replace(r'Bangalore', 'Bengaluru', regex=True)
df['Location'] = df['Location'].str.replace(r'Bhubneswar', 'Bhubaneswar', regex=Tru
df['Location'] = df['Location'].str.replace(r'Kolkatta', 'Kolkata', regex=True)
df['Location'] = df['Location'].str.replace(r'Nw Delhi', 'New Delhi', regex=True)
df['Location'] = df['Location'].str.replace(r'\bUS\b', 'USA', regex=True)
df['Location'] = df['Location'].str.replace(r'\\\\xc2\\\\xa0', '', regex=True)
df['Location'] = df['Location'].fillna('')  # Fill NaN values with an empty string
df['Location'] = df['Location'].replace({'Ahemadabad': 'Ahmedabad', 'Ahemdabad': 'A
df.loc[df['Location'].str.contains('/'), 'Location'] = 'Multiple Cities'
df.loc[df['Location'].str.contains('&'), 'Location'] = 'Multiple Cities'
df.loc[df['Location'].str.contains('and'), 'Location'] = 'Multiple Cities'
df.loc[df['Location'].str.contains(','), 'Location'] = 'Multiple Cities'
```

In [ ]:
```
# show unique values in city column
print(df["Location"].nunique())
df["Location"].unique()
```

66

Out[ ]:
```
array(['Bengaluru', 'Gurgaon', 'New Delhi', 'Mumbai', 'Chennai', 'Pune',
       'Noida', 'Faridabad', 'San Francisco', 'Multiple Cities',
       'Amritsar', 'Delhi', 'Kormangala', 'Tulangan', 'Hyderabad',
       'Burnsville', 'Menlo Park', 'Gurugram', 'Palo Alto',
       'Santa Monica', 'Singapore', 'Taramani', 'Andheri', 'Chembur',
       'Nairobi', 'Haryana', 'New York', 'Karnataka', 'Bhopal', 'Jaipur',
       'Nagpur', 'Indore', 'California', 'India', 'Ahmedabad', 'Rourkela',
       'Srinagar', 'Bhubaneswar', 'Kolkata', 'Coimbatore', 'Udaipur', '',
       'Surat', 'Goa', 'Uttar Pradesh', 'Gaya', 'Vadodara', 'Missourie',
       'Panaji', 'Gwalior', 'Karur', 'Udupi', 'Kochi', 'Agra', 'Hubli',
       'Kerala', 'Kozhikode', 'USA', 'Siliguri', 'Lucknow', 'Kanpur',
       'London', 'Varanasi', 'Jodhpur', 'Boston', 'Belgaum'], dtype=object)
```

In [ ]:
```
# remove duplicate rows
df.drop_duplicates(inplace=True)
# cleaning Startup Name column
df["Startup Name"] = df["Startup Name"].str.replace(r"https://www.wealthbucket.in/"
df["Startup Name"] = df["Startup Name"].str.replace(r"\\", "", regex=True)
df["Startup Name"] = df["Startup Name"].str.replace(r"xc2xa0", "", regex=True)
df["Startup Name"] = df["Startup Name"].str.replace(r"xe2x80x99", "'", regex=True)
df.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	Am
0	09/01/2020	BYJU'S	E-Tech	Bengaluru	Tiger Global Management	Private Equity Round	20,0
1	13/01/2020	Shuttl	Transportation	Gurgaon	Susquehanna Growth Equity	Series C	8
2	09/01/2020	Mamaearth	E-commerce	Bengaluru	Sequoia Capital India	Series B	1,8
3	02/01/2020	Wealth Bucket	FinTech	New Delhi	Vinod Khatumal	Pre-series A	3
4	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	1

In [ ]:
```
# cleaning Amount in USD column
df["Amount in USD"] = df["Amount in USD"].str.replace(r"\\\\xc2\\\\xa0", "", regex=
df["Amount in USD"] = df["Amount in USD"].str.replace(r"\\xc2\\xa0", "", regex=True
df["Amount in USD"] = df["Amount in USD"].str.replace(r"unknown", "undisclosed", re
df["Amount in USD"] = df["Amount in USD"].str.replace(r"Undisclosed", "undisclosed"
df["Amount in USD"] = df["Amount in USD"].str.replace(r"N/A", "undisclosed", regex=
df["Amount in USD"] = df["Amount in USD"].str.replace(r"\+", "", regex=True)
df.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	Am
0	09/01/2020	BYJU'S	E-Tech	Bengaluru	Tiger Global Management	Private Equity Round	20,0
1	13/01/2020	Shuttl	Transportation	Gurgaon	Susquehanna Growth Equity	Series C	8
2	09/01/2020	Mamaearth	E-commerce	Bengaluru	Sequoia Capital India	Series B	1,8
3	02/01/2020	Wealth Bucket	FinTech	New Delhi	Vinod Khatumal	Pre-series A	3
4	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	1

In [ ]:
```
# replace blank values with NaN
df.replace('', np.nan, inplace=True)
# remove rows with more than 2 NaN values
df.dropna(thresh=5, inplace=True)
df.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	Am
0	09/01/2020	BYJU'S	E-Tech	Bengaluru	Tiger Global Management	Private Equity Round	20,0
1	13/01/2020	Shuttl	Transportation	Gurgaon	Susquehanna Growth Equity	Series C	8
2	09/01/2020	Mamaearth	E-commerce	Bengaluru	Sequoia Capital India	Series B	1,8
3	02/01/2020	Wealth Bucket	FinTech	New Delhi	Vinod Khatumal	Pre-series A	3
4	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	1

In [ ]:
```
# reset index
df.reset_index(drop=True, inplace=True)
df.tail()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	
2999	28/01/2015	Grabhouse.com	NaN	NaN	Kalaari Capital, Sequoia Capital	Private Equity	2
3000	29/01/2015	Printvenue	NaN	NaN	Asia Pacific Internet Group	Private Equity	4
3001	29/01/2015	Graphene	NaN	NaN	KARSEMVEN Fund	Private Equity	
3002	30/01/2015	Mad Street Den	NaN	NaN	Exfinity Fund, GrowX Ventures.	Private Equity	1
3003	31/01/2015	couponmachine.in	NaN	NaN	UK based Group of Angel Investors	Seed Funding	

In [ ]:
```
# randomly fill NaN values in location column with either Bengaluru, Mumbai, New De
df["Location"].fillna(pd.Series(np.random.choice(["Bengaluru", "Mumbai", "New Delhi
# fill NaN values in Industry Vertical column with Unknown
df["Industry Vertical"].fillna("Unknown", inplace=True)
df.tail()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType
2999	28/01/2015	Grabhouse.com	Unknown	Bengaluru	Kalaari Capital, Sequoia Capital	Private Equity
3000	29/01/2015	Printvenue	Unknown	Mumbai	Asia Pacific Internet Group	Private Equity
3001	29/01/2015	Graphene	Unknown	Mumbai	KARSEMVEN Fund	Private Equity
3002	30/01/2015	Mad Street Den	Unknown	Bengaluru	Exfinity Fund, GrowX Ventures.	Private Equity
3003	31/01/2015	couponmachine.in	Unknown	Bengaluru	UK based Group of Angel Investors	Seed Funding

In [ ]:
```
# fill NaN values in Investors Name column with Unknown
df["Investors Name"].fillna("Unknown", inplace=True)
# randomly fill NaN values in Investment Type column with either Seed Funding or Pr
df["InvestmentnType"].fillna(pd.Series(np.random.choice(["Seed Funding", "Private E
# fill nan values in Amount in USD column with undisclosed
df["Amount in USD"].fillna("undisclosed", inplace=True)
df.tail()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType
2999	28/01/2015	Grabhouse.com	Unknown	Bengaluru	Kalaari Capital, Sequoia Capital	Private Equity
3000	29/01/2015	Printvenue	Unknown	Mumbai	Asia Pacific Internet Group	Private Equity
3001	29/01/2015	Graphene	Unknown	Mumbai	KARSEMVEN Fund	Private Equity
3002	30/01/2015	Mad Street Den	Unknown	Bengaluru	Exfinity Fund, GrowX Ventures.	Private Equity
3003	31/01/2015	couponmachine.in	Unknown	Bengaluru	UK based Group of Angel Investors	Seed Funding

In [ ]:
```
# check for null values
df.isnull().sum()
```

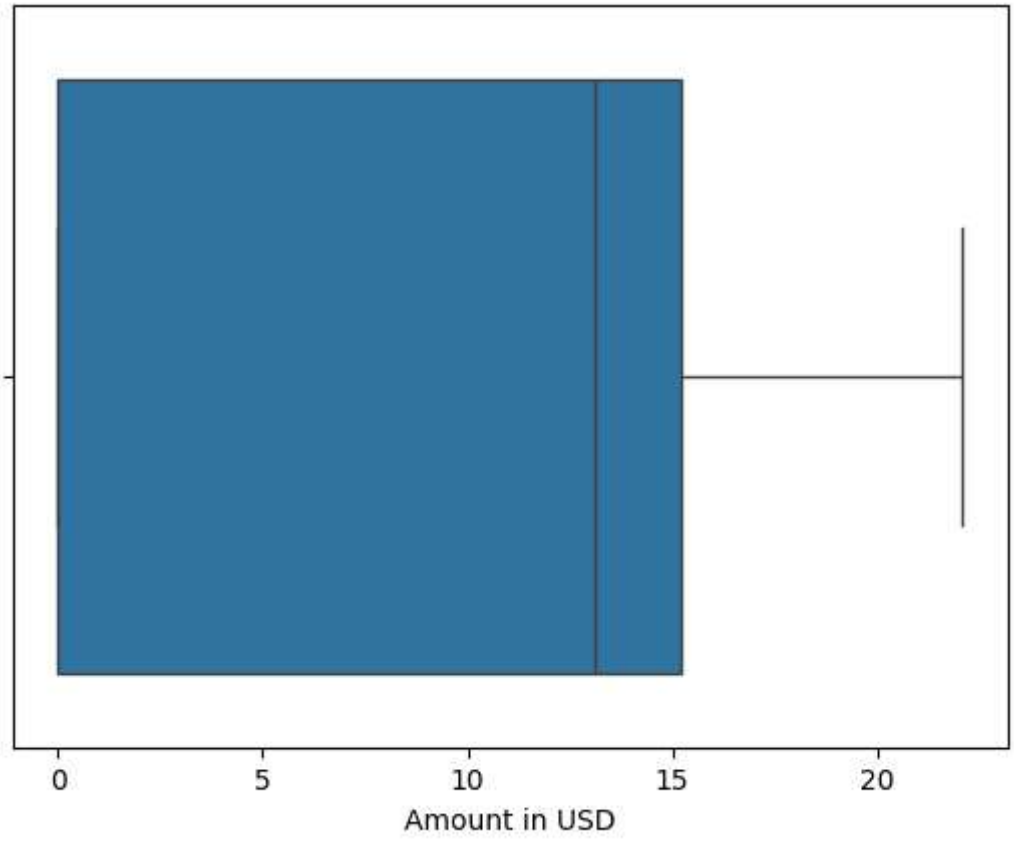
Out[ ]:
```
Date dd/mm/yyyy      0
Startup Name         0
Industry Vertical    0
Location             0
Investors Name       0
InvestmentnType      0
Amount in USD        0
dtype: int64
```

In [ ]:
```
# transform variables using log transformation
df["Amount in USD"] = df["Amount in USD"].str.replace(r",", "", regex=True)
df["Amount in USD"] = df["Amount in USD"].str.replace(r"undisclosed", "1", regex=Tr
df["Amount in USD"] = df["Amount in USD"].astype("float64")
df["Amount in USD"] = np.log(df["Amount in USD"])
df.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	An ir
0	09/01/2020	BYJU'S	E-Tech	Bengaluru	Tiger Global Management	Private Equity Round	19.1
1	13/01/2020	Shuttl	Transportation	Gurgaon	Susquehanna Growth Equity	Series C	15.9
2	09/01/2020	Mamaearth	E-commerce	Bengaluru	Sequoia Capital India	Series B	16.7
3	02/01/2020	Wealth Bucket	FinTech	New Delhi	Vinod Khatumal	Pre-series A	14.9
4	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	14.4

In [ ]:
```
# Box plot
sns.boxplot(x=df["Amount in USD"])
plt.show()
```



In [ ]:
```
df.describe()
```

Out[ ]:

	Amount in USD
count	3004.000000
mean	10.008408
std	6.911940
min	0.000000
25%	0.000000
50%	13.122363
75%	15.201805
max	22.084242

RESULT	**Data Cleaning and Preprocessing:** There are no missing values in the dataset. All values have been properly imputed.
CONCLUSION	In this experiment we learned how to handle missing values by imputing them with mode. We also learned how to check for outliers and how to transform variables using log transformation. We learned how to remove duplicate records from a data frame. We also learned how to standardize date formats across our dataset.