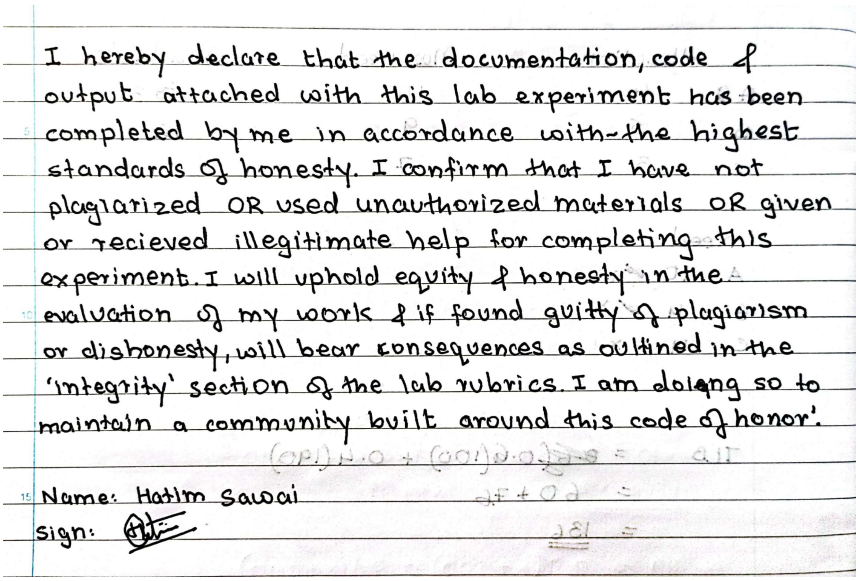
Name	Hatim Sawai
UID No.	2021300108

Experiment 3

HONOR PLEDGE	
PROBLEM STATEMENT	**Data Integration and Reshaping:** 1. Merge two or more data frames based on a common key: Create a new pandas dataframe with containing 20 records and 5 attributes. One attribute should compulsorily be a categorical variable, which is common with a categorical attribute from the CSV dataset that has been used earlier by you. The other 4 attributes can be generated on reasonable assumptions. Merge these 2 datasets on the common key 2. Concatenate multiple Data Frames vertically or horizontally: Create 5 new rows of the same schema as the original csv dataframe. Use a new categorical value for the common key attribute. Concatenate this horizontally with the existing dataframe Similarly, execute a vertical concatenation with a mock dataframe. 3. Pivot a Data Frame from long to wide format or vice versa: Add reasoning for using pivot. Explain with a relevant example how pivot operation is useful in data analysis 4. Stack and unstack columns or levels in the Data Frame: Reason about the use and application of stacking and

	unstacking with the help of the current dataframe or another example. 5. Data Wrangling: Experiment with other techniques in data wrangling to convert and reshape your dataframe into its final state which can be used for analysis
THEORY	1. **Merging Data Frames** Data frames often hold data from separate sources, each with its own valuable insights. Merging bridges this gap, allowing you to combine them based on shared identifiers. Imagine analyzing customer purchase history. One data frame might hold customer demographics, another their transactions. Merging lets you explore how demographics influence purchase behavior, painting a holistic picture. **For this Dataset:** I have merged the original dataframe with a new dataframe with 5 new attributes: location, state, revenue, no. of employees & product_type. I have performed an inner join on the common key 'Location' to merge the 2 dataframes. 2. **Concatenation of Data Frames** Ans: Sometimes, adding data means expanding dimensions. Concatenation empowers you to do just that, either vertically or horizontally. Vertical concatenation, stacking data frames like building blocks, increases the number of observations. Picture analyzing sales data across multiple stores. Stacking monthly sales data from each store creates a comprehensive timeline for analysis. Horizontal concatenation, joining data frames side-by-side, adds new features. Think about adding weather data to sales data. Concatenation horizontally creates a richer dataset for exploring sales-weather relationships. **For this Dataset:** I have concatenated the dataframes both horizontally and vertically. I have used extra 5 rows of the same schema as the original csv dataframe with "Location: Jamnagar" as a new attribute value for the horizontal concatenation. I have used a new Remarks column for the vertical concatenation. 3. **Pivoting Data Frames** Pivot tables are data transformers, summarizing and reorganizing data from a wide format to a more condensed and insightful one. Imagine a vast sales dataset with rows for each transaction, listing product, customer, price, and quantity. A pivot table can group by product and customer, calculating quantities sold and average prices. This condensed view instantly reveals top-selling products for each customer, uncovering valuable buying patterns.

For this Dataset:
I have pivoted the dataframe from long to wide format. I have used the pivot_table function to pivot the dataframe. I have used the 'Location' as the index, 'Remarks' as the columns and 'Amount in USD' as the values.

4. **Stacking/Unstacking Data Frames**
Data often needs reshaping for specific analysis needs. Stacking and unstacking are powerful tools for this. Stacking takes wide data (many columns) and condenses it into long data (fewer columns with repeated values). Imagine a dataset with city, year, and separate sales figures for each product. Stacking creates rows for each city-year combination, with columns for product and sales, making it easier to analyze trends across products and time. Unstacking reverses this, going from long to wide format. It might be needed for heatmaps or visualizations that require product-specific sales figures for each city-year.

For this Dataset:
I have stacked the dataframe to convert it from wide to long format. I have used the stack() function to stack the dataframe. I have also unstacked the dataframe to convert it from long to wide format. I have used the unstack() function to unstack the dataframe.

5. **Data Wrangling**
Data wrangling is the often invisible but crucial first step in any data analysis journey. It's the art of cleaning, transforming, and organizing raw data into a usable format. Imagine a messy room filled with scattered information. Data wrangling is like decluttering and organizing that room, making it easy to navigate and find what you need. It removes inconsistencies, fixes errors, and formats data appropriately, ensuring the analysis is based on accurate and reliable information. This saves time and effort later, as you're working with clean, ready-to-use data, ultimately leading to more trustworthy and impactful insights.

For this Dataset:
Additonally, I have used other data wrangling functions like: groupby(), mean(), sort_values() to get some analysis on the dataframe like location with highest average investment.

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/cleaned_startup_funding.csv')
df.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	An i
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

2. Merging data frames based on a common key

In [ ]:
```
new_data = {
    'Location': np.random.choice(['Bangalore', 'Mumbai', 'Delhi', 'Chennai', 'Hyder
    'Revenue': np.random.randint(1000000, 9999999, 20),
    'Employees': np.random.randint(100, 999, 20),
    'product_type': np.random.choice(['Software', 'Hardware'], size=20),
}
df2 = pd.DataFrame(new_data)
df2['State'] = df2['Location'].map({
    'Bangalore': 'Karnataka',
    'Mumbai': 'Maharashtra',
    'Delhi': 'NCR',
    'Chennai': 'Tamil Nadu',
    'Hyderabad': 'Telangana'
})
df2.head()
```

Out[ ]:

	Location	Revenue	Employees	product_type	State
0	Hyderabad	7436410	165	Hardware	Telangana
1	Bangalore	9495687	775	Software	Karnataka
2	Chennai	2623091	350	Hardware	Tamil Nadu
3	Chennai	6750091	910	Hardware	Tamil Nadu
4	Bangalore	5538119	910	Hardware	Karnataka

In [ ]:
```
# perform inner join on df and df2
df3 = pd.merge(df, df2, on='Location', how='inner')
df3.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	Amount in USD	Reve
0	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	14.403297	8154
1	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	14.403297	6510
2	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	14.403297	1771
3	02/01/2020	Fashor	Fashion and Apparel	Mumbai	Sprout Venture Partners	Seed Round	14.403297	7457
4	16/12/2019	InCred	Finance	Mumbai	Unknown	Debt Funding	15.590463	8154

3. Concatenating multiple Data Frames vertically/horizontally

In [ ]:
```
# create 5 new rows for original dataframe with location value as Jamnagar
extra_data = {
    'Date dd/mm/yyyy': np.random.choice(['01/01/2017', '01/02/2017', '01/03/2017',
    'Startup Name': ['Udemy', 'Reliance', 'IDBI Bank', 'Blinkit', 'Pepsico'],
    'Industry Vertical': ['EduTech', 'Telecom', 'Banking', 'IT', 'Food & Beverage']
    'Location': np.repeat('Jamnagar', 5), # changed attribute value
    'Investors Name': ['Softbank', 'Alibaba', 'Tencent', 'Sequoia', 'Accel Partners
    'InvestmentnType': np.random.choice(['Seed Funding', 'Private Equity'], 5),
    'Amount in USD': np.random.randint(1000000, 9999999, 5),
}
df4 = pd.DataFrame(extra_data)
df4['Amount in USD'] = np.log(df4['Amount in USD'])
df4.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	Amount in USD
0	01/04/2017	Udemy	EduTech	Jamnagar	Softbank	Seed Funding	16.040169
1	01/05/2017	Reliance	Telecom	Jamnagar	Alibaba	Private Equity	15.149043
2	01/05/2017	IDBI Bank	Banking	Jamnagar	Tencent	Private Equity	14.967510
3	01/03/2017	Blinkit	IT	Jamnagar	Sequoia	Seed Funding	15.229681
4	01/01/2017	Pepsico	Food & Beverage	Jamnagar	Accel Partners	Private Equity	16.091184

In [ ]:
```
# concatenate df and df4, horizontally
df5 = pd.concat([df, df4], axis=0)
df5.tail()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	Amount in USD
0	01/04/2017	Udemy	EduTech	Jamnagar	Softbank	Seed Funding	16.040169
1	01/05/2017	Reliance	Telecom	Jamnagar	Alibaba	Private Equity	15.149043
2	01/05/2017	IDBI Bank	Banking	Jamnagar	Tencent	Private Equity	14.967510
3	01/03/2017	Blinkit	IT	Jamnagar	Sequoia	Seed Funding	15.229681
4	01/01/2017	Pepsico	Food & Beverage	Jamnagar	Accel Partners	Private Equity	16.091184

In [ ]:
```
# creating new df with remarks column with good, average or badn values
df6 = pd.DataFrame({
    'Remarks': np.random.choice(['Good', 'Average', 'Bad'], size=df5.shape[0])
```

```
})
df6.head()
```

Out[ ]:

	Remarks
0	Bad
1	Good
2	Good
3	Bad
4	Bad

In [ ]:
```
# concatenate df5 and df6, vertically
df5.reset_index(drop=True, inplace=True)
df7 = pd.concat([df5, df6], axis=1)
df7.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	An i
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

4. Pivoting a Data Frame from long to wide format or vice versa

In [ ]:
```
# pivoting df7 with Location as index
df8 = df7.pivot_table(index='Location', columns='Remarks', values='Amount in USD',
df8.head()
```

Out[ ]:

Remarks	Average	Bad	Good
Location			
Agra	NaN	0.000000	0.000000
Ahmedabad	8.851581	10.729219	10.332156
Amritsar	12.611538	NaN	NaN
Andheri	15.564710	NaN	NaN
Belgaum	13.122363	NaN	NaN

5. Stacking & unstacking columns/levels in the Data Frame

In [ ]:
```
# stacking df
df9 = df.stack()
df9.head()
```

Out[ ]:
```
0  Date dd/mm/yyyy                  09/01/2020
   Startup Name                         BYJU'S
   Industry Vertical                    E-Tech
   Location                           Bengaluru
   Investors Name       Tiger Global Management
dtype: object
```

In [ ]:
```
# unstacking df
df10 = df9.unstack()
df10.head()
```

Out[ ]:

	Date dd/mm/yyyy	Startup Name	Industry Vertical	Location	Investors Name	InvestmentnType	An i
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

6. Data Wrangling

In [ ]:
```
df11 = df.copy()
# reversing log transformation on Amount in USD column
df11['Amount in USD'] = np.exp(df11['Amount in USD'])
# displaying the mean of Amount in USD for each location in descending order
df11 = df11.groupby('Location')['Amount in USD'].mean().sort_values(ascending=False
df11['Amount in USD'] = df11['Amount in USD'].round(2).astype(int)
df11.head()
```

Out[ ]:

	Location	Amount in USD
0	Menlo Park	450000000
1	California	300000000
2	Tulangan	200000000
3	Kormangala	142000000
4	Santa Monica	110000000

CONCLUSION	In this experiment we learned how to merge 2 data frames using a common key, how to concatenate them either horizontally or vertically. Also we learned how to pivot a dataframe or stack/unstack it. Lastly we implemented extra data wrangling functions like groupby, sort_values, etc. to get some analysis on the dataframe.