Project Title: Data Preparation and Visualization for Business Metrics

```
#Setting up and Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Loading dataset
file_path = "/content/P3-Future-500-The-Dataset.csv"
fin = pd.read_csv(file_path, na_values=[""])
```

```
# Displaying the first few rows of the dataframe
fin.head()
```

	ID	Name	Industry	Inception	Employees	State	City	Revenue	Expenses	Profit	Growth
0	1	Over-Hex	Software	2006.0	25.0	TN	Franklin	$9,684,527	1,130,700 Dollars	8553827.0	19%
1	2	Unimattax	IT Services	2009.0	36.0	PA	Newtown Square	$14,016,543	804,035 Dollars	13212508.0	20%
2	3	Greenfax	Retail	2012.0	NaN	SC	Greenville	$9,746,272	1,044,375 Dollars	8701897.0	16%
3	4	Blacklane	IT Services	2011.0	66.0	CA	Orange	$15,359,369	4,631,808 Dollars	10727561.0	19%
4	5	Yearflex	Software	2013.0	45.0	WI	Madison	$8,567,910	4,374,841 Dollars	4193069.0	19%

Next steps: View recommended plots New interactive sheet

```
#Understanding the structure and data quality
#Class Information
#Index Range
#Data Columns
#Data Types Summary
#Memory Usage

fin.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   ID          500 non-null    int64
 1   Name        500 non-null    object
 2   Industry    498 non-null    object
 3   Inception   499 non-null    float64
 4   Employees   498 non-null    float64
 5   State       496 non-null    object
 6   City        500 non-null    object
 7   Revenue     498 non-null    object
 8   Expenses    497 non-null    object
 9   Profit      498 non-null    float64
 10  Growth      499 non-null    object
dtypes: float64(3), int64(1), object(7)
memory usage: 43.1+ KB
```

The dataframe consists of 500 entries and 11 columns, including information about ID, Name, Industry, financials, and location. It has a few missing values in the 'Industry', 'Employees', 'State', 'Revenue', and 'Expenses' columns.

'Industry': 2 missing values

'Employees': 2 missing values

'State': 4 missing values

'Revenue': 2 missing values

'Expenses': 3 missing values

Data Cleaning and Transformation

```
#Removing unwanted characters from 'Expenses', 'Revenue', and 'Growth' columns
fin['Expenses'] = fin['Expenses'].astype(str).str.replace(' Dollars', '').str.replace(',', '').astype(float)
fin['Revenue'] = fin['Revenue'].astype(str).str.replace(r'\$', '', regex=True).str.replace(',', '').astype(f
fin['Growth'] = fin['Growth'].astype(str).str.replace('%', '').astype(float)

# Displaying the structure of the cleaned dataframe
fin.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ID         500 non-null    int64
 1   Name       500 non-null    object
 2   Industry   498 non-null    object
 3   Inception  499 non-null    float64
 4   Employees  498 non-null    float64
 5   State      496 non-null    object
 6   City       500 non-null    object
 7   Revenue    498 non-null    float64
 8   Expenses   497 non-null    float64
 9   Profit     498 non-null    float64
 10  Growth     499 non-null    float64
dtypes: float64(6), int64(1), object(4)
memory usage: 43.1+ KB
```

The Expenses, Revenue, and Growth columns have been cleaned by removing unwanted characters and converting them to float type.

The Expenses column had 'Dollars' and commas removed; the Revenue column had dollar signs and commas removed; and the Growth column had percentage signs removed.

The cleaned DataFrame now has numeric values in these columns and is ready for further analysis.

Handling Missing Data

```
# Replace missing 'State' based on 'City'
fin.loc[(fin['City'] == 'New York') & (fin['State'].isnull()), 'State'] = 'NY'
fin.loc[(fin['City'] == 'San Francisco') & (fin['State'].isnull()), 'State'] = 'CA'

# Impute missing values using median for each industry
fin['Employees'] = fin.groupby('Industry')['Employees'].transform(lambda x: x.fillna(x.median()))
fin['Growth'] = fin.groupby('Industry')['Growth'].transform(lambda x: x.fillna(x.median()))
fin['Revenue'] = fin.groupby('Industry')['Revenue'].transform(lambda x: x.fillna(x.median()))
fin['Expenses'] = fin.groupby('Industry')['Expenses'].transform(lambda x: x.fillna(x.median()))

# Derive missing 'Profit' values
fin['Profit'] = fin['Revenue'] - fin['Expenses']

# Check for any remaining missing values
print(fin.isnull().sum())
```

```
ID           0
Name         0
Industry     2
Inception    1
Employees    2
State        0
City         0
Revenue      2
Expenses     2
Profit       2
Growth       2
dtype: int64
```

State Replacement: Filled missing 'State' values with 'NY' for 'New York' and 'CA' for 'San Francisco'.

Imputation of Missing Values: Replaced missing values in 'Employees', 'Growth', 'Revenue', and 'Expenses' with the median for each 'Industry'.

Derivation of Missing 'Profit' Values: Computed 'Profit' as the difference between 'Revenue' and 'Expenses'.

Verification: Checked for remaining missing values with print(fin.isnull().sum()).

These steps address missing data issues and ensure the 'Profit' column is accurately calculated.

```
fin.head()
```

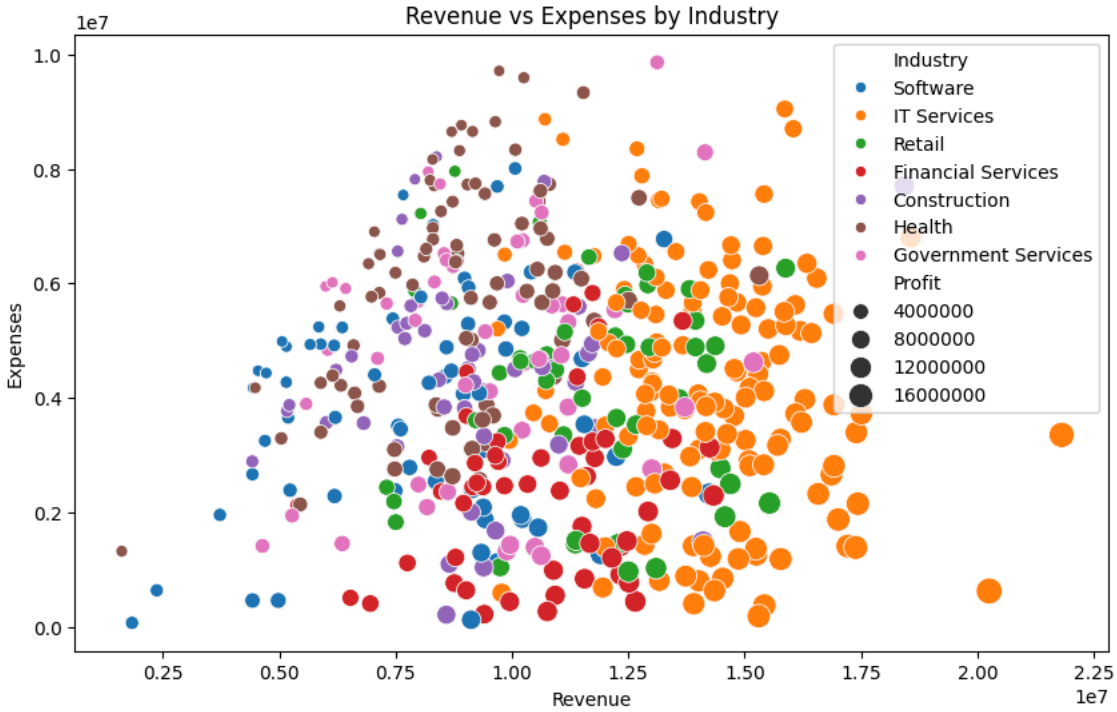
	ID	Name	Industry	Inception	Employees	State	City	Revenue	Expenses	Profit	Growth
0	1	Over-Hex	Software	2006.0	25.0	TN	Franklin	9684527.0	1130700.0	8553827.0	19.0
1	2	Unimattax	IT Services	2009.0	36.0	PA	Newtown Square	14016543.0	804035.0	13212508.0	20.0
2	3	Greenfax	Retail	2012.0	28.0	SC	Greenville	9746272.0	1044375.0	8701897.0	16.0
3	4	Blacklane	IT Services	2011.0	66.0	CA	Orange	15359369.0	4631808.0	10727561.0	19.0
4	5	Yearflex	Software	2013.0	45.0	WI	Madison	8567910.0	4374841.0	4193069.0	19.0

Next steps: View recommended plots New interactive sheet

Data Visualization

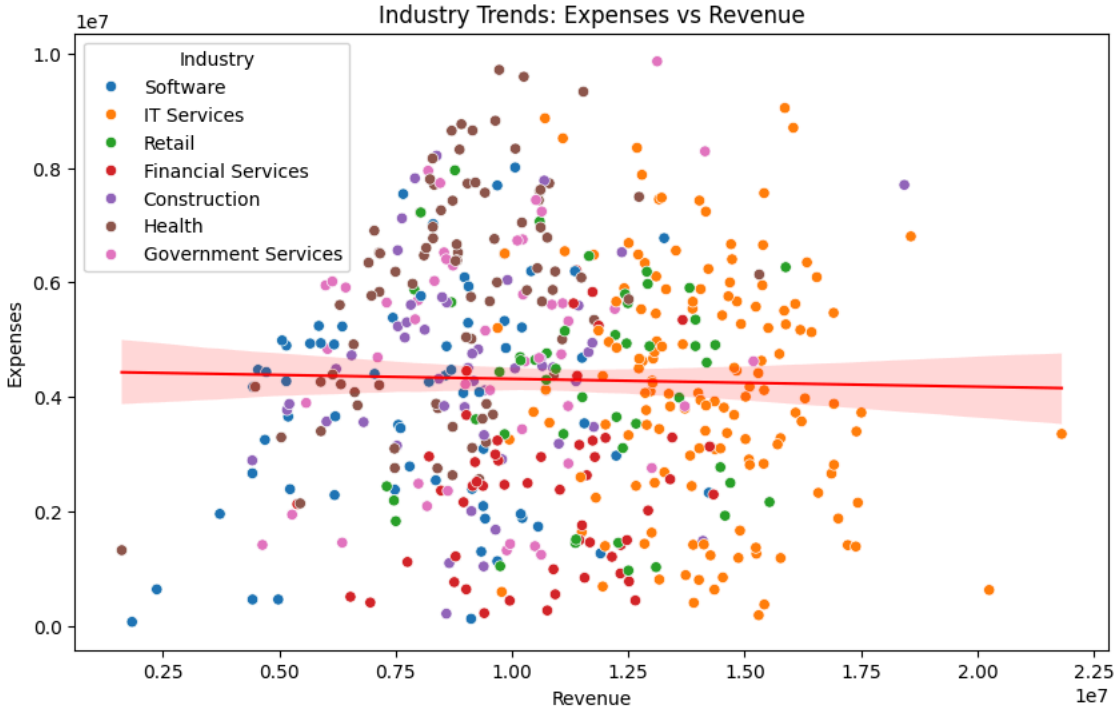
```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
#Challenge 1: Scatterplot Classified by Industry (Revenue, Expenses, and Profit)

plt.figure(figsize=(10, 6))
sns.scatterplot(data=fin, x='Revenue', y='Expenses', hue='Industry', size='Profit', sizes=(40, 200))
plt.title('Revenue vs Expenses by Industry')
plt.show()
```



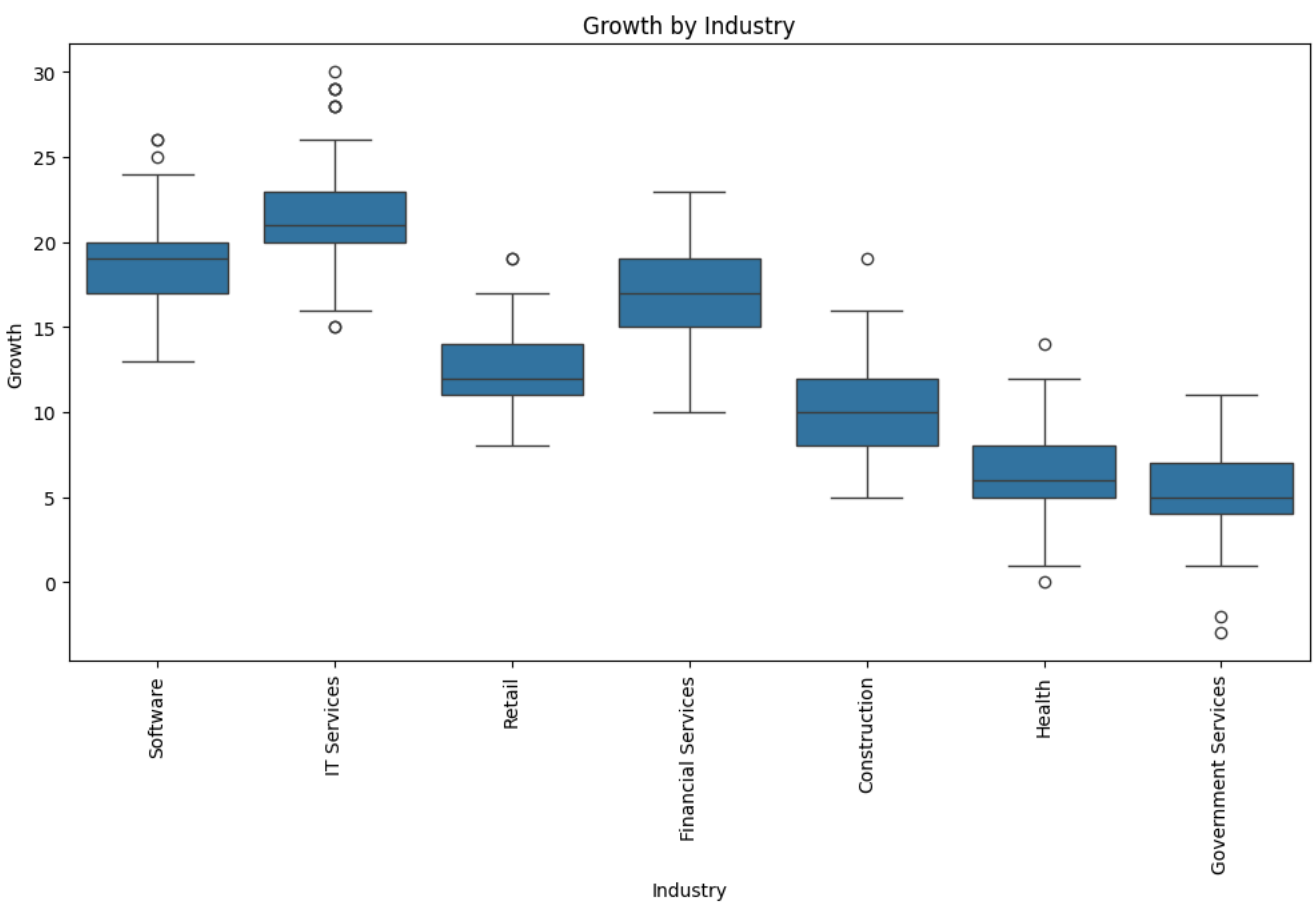
Industries like Technology and Financial Services have higher revenue and expenses and are associated with higher profits. Retail and Construction tend to operate with lower revenues and profits, indicating smaller-scale operations.

```
#Challenge 2: Industry Trends for Expenses vs Revenue

plt.figure(figsize=(10, 6))
sns.scatterplot(data=fin, x='Revenue', y='Expenses', hue='Industry')
sns.regplot(data=fin, x='Revenue', y='Expenses', scatter=False, color='red', line_kws={"linewidth": 1.5})
plt.title('Industry Trends: Expenses vs Revenue')
plt.show()
```



Industries such as Technology and Financial Services show a strong correlation between expenses and revenue, indicating that higher revenue results in higher expenses. On the other hand, Retail and Construction show less correlation, suggesting that these industries may manage their costs more effectively as they grow.

```
#Challenge 3: BoxPlot Showing Growth by Industry

plt.figure(figsize=(12, 6))
sns.boxplot(data=fin, x='Industry', y='Growth')
plt.title('Growth by Industry')
plt.xticks(rotation=90)
plt.show()
```



Technology and Financial Services show higher growth variability but potential for high growth, while Construction and Retail show more consistent, lower growth.