

# WALMART FINANCIAL ANALYSIS 2012 - 2022

## Importing relevant libraries

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sb  
import warnings
```

```
In [2]: ## Notebook setup
```

```
In [3]: warnings.filterwarnings("ignore")
```

## Loading the dataset

```
In [4]: df = pd.read_csv("Financials Sample Data.csv")
```

## Data Exploration

```
In [5]: df.head()
```

```
Out[5]:
```

	Account	Business Unit	Currency	Year	Scenario	Jan	Feb	Mar	Apr
0	Sales	Software	USD	2012	Actuals	\$90,924,002	\$82,606,134	\$72,780,220	\$52,943,701
1	Cost of Goods Sold	Software	USD	2012	Actuals	(\$41,623,278)	(\$40,464,347)	(\$30,806,326)	(\$21,412,962)
2	Commissions Expense	Software	USD	2012	Actuals	(\$4,454,359)	(\$3,386,032)	(\$3,389,705)	(\$2,149,257)
3	Payroll Expense	Software	USD	2012	Actuals	(\$9,901,680)	(\$9,871,172)	(\$8,459,696)	(\$6,303,408)
4	Travel & Entertainment Expense	Software	USD	2012	Actuals	(\$951,255)	(\$838,985)	(\$872,700)	(\$624,416)

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 351 entries, 0 to 350
Data columns (total 17 columns):
 #   Column      Non-Null Count Dtype
 ---  -----
 0   Account     351 non-null    object
 1   Business Unit 351 non-null    object
 2   Currency    351 non-null    object
 3   Year        351 non-null    int64
 4   Scenario    351 non-null    object
 5   Jan         351 non-null    object
 6   Feb         351 non-null    object
 7   Mar         351 non-null    object
 8   Apr         351 non-null    object
 9   May         351 non-null    object
 10  Jun         351 non-null    object
 11  Jul         351 non-null    object
 12  Aug         351 non-null    object
 13  Sep         351 non-null    object
 14  Oct         351 non-null    object
 15  Nov         351 non-null    object
 16  Dec         351 non-null    object
dtypes: int64(1), object(16)
memory usage: 46.7+ KB
```

In [7]: `df.shape`

Out[7]: (351, 17)

## Data Cleaning

```
In [8]: for col in df.columns[5:]:
    df[col] = df[col].str.replace('$', '')
    df[col] = df[col].str.replace('(', '')
    df[col] = df[col].str.replace(')', '')
    df[col] = df[col].str.replace(',', '')
    df[col] = df[col].astype(int)
```

In [9]: `df.head()`

	Account	Business Unit	Currency	Year	Scenario	Jan	Feb	Mar	Apr	May
0	Sales	Software	USD	2012	Actuals	90924002	82606134	72780220	52943701	77528109
1	Cost of Goods Sold	Software	USD	2012	Actuals	41623278	40464347	30806326	21412962	37047252
2	Commissions Expense	Software	USD	2012	Actuals	4454359	3386032	3389705	2149257	3168079
3	Payroll Expense	Software	USD	2012	Actuals	9901680	9871172	8459696	6303408	8493573
4	Travel & Entertainment Expense	Software	USD	2012	Actuals	951255	838985	872700	624416	919835

In [10]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 351 entries, 0 to 350
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Account          351 non-null    object  
 1   Business Unit    351 non-null    object  
 2   Currency          351 non-null    object  
 3   Year              351 non-null    int64  
 4   Scenario          351 non-null    object  
 5   Jan               351 non-null    int32  
 6   Feb               351 non-null    int32  
 7   Mar               351 non-null    int32  
 8   Apr               351 non-null    int32  
 9   May               351 non-null    int32  
 10  Jun               351 non-null    int32  
 11  Jul               351 non-null    int32  
 12  Aug               351 non-null    int32  
 13  Sep               351 non-null    int32  
 14  Oct               351 non-null    int32  
 15  Nov               351 non-null    int32  
 16  Dec               351 non-null    int32  
dtypes: int32(12), int64(1), object(4)
memory usage: 30.3+ KB
```

In [11]: `df.to_csv('clean_financial_data.csv')`

In [12]: `df.head()`

Out[12]:

	Account	Business Unit	Currency	Year	Scenario	Jan	Feb	Mar	Apr	May
<b>0</b>	Sales	Software	USD	2012	Actuals	90924002	82606134	72780220	52943701	77528109
<b>1</b>	Cost of Goods Sold	Software	USD	2012	Actuals	41623278	40464347	30806326	21412962	37047252
<b>2</b>	Commissions Expense	Software	USD	2012	Actuals	4454359	3386032	3389705	2149257	3168079
<b>3</b>	Payroll Expense	Software	USD	2012	Actuals	9901680	9871172	8459696	6303408	8493573
<b>4</b>	Travel & Entertainment Expense	Software	USD	2012	Actuals	951255	838985	872700	624416	919835

## Data Transformation

In [13]: `df['Total'] = np.array(df[df.columns[5:]].sum(axis=1))`

In [14]: `df.head()`

Out[14]:

	Account	Business Unit	Currency	Year	Scenario	Jan	Feb	Mar	Apr	May
0	Sales	Software	USD	2012	Actuals	90924002	82606134	72780220	52943701	77528109
1	Cost of Goods Sold	Software	USD	2012	Actuals	41623278	40464347	30806326	21412962	37047252
2	Commissions Expense	Software	USD	2012	Actuals	4454359	3386032	3389705	2149257	3168079
3	Payroll Expense	Software	USD	2012	Actuals	9901680	9871172	8459696	6303408	8493573
4	Travel & Entertainment Expense	Software	USD	2012	Actuals	951255	838985	872700	624416	919835

In [15]: df.Year.unique()

Out[15]: array([2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], dtype=int64)

In [16]: df = df[df['Year'] != 2023]

In [17]: df.Year.unique()

Out[17]: array([2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022], dtype=int64)

In [18]: df.to\_csv('financial data 2012 - 2022.csv')

## Sales Analysis and Visualization

In [19]: sales = df[df['Account'] == 'Sales']

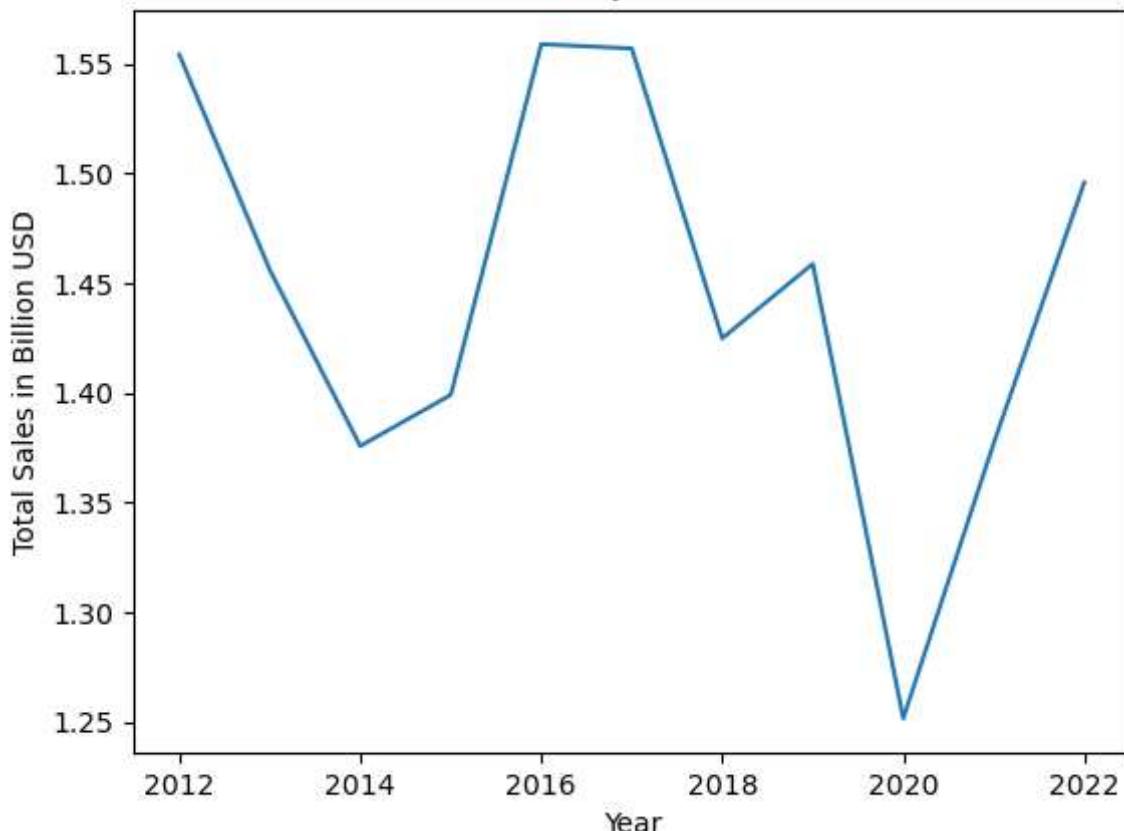
In [20]: total\_sales\_by\_year = sales.groupby('Year')['Total'].sum() / 1\_000\_000\_000

In [21]: total\_sales\_by\_year.sort\_values(ascending=False)

```
Out[21]:
Year
2016    1.558807
2017    1.556849
2012    1.554061
2022    1.495748
2019    1.458740
2013    1.455984
2018    1.424819
2015    1.399039
2021    1.377023
2014    1.375727
2020    1.251657
Name: Total, dtype: float64
```

```
In [22]: total_sales_by_year.plot()
plt.title("Walmart Total Sales by Year from 2012 - 2022")
plt.xlabel('Year')
plt.ylabel('Total Sales in Billion USD')
plt.savefig('sales.png')
plt.show()
```

### Walmart Total Sales by Year from 2012 - 2022



```
In [23]: total_sales_by_year.sort_values(ascending=False)
```

```
Out[23]:
```

Year	Total Sales in Billion USD
2016	1.558807
2017	1.556849
2012	1.554061
2022	1.495748
2019	1.458740
2013	1.455984
2018	1.424819
2015	1.399039
2021	1.377023
2014	1.375727
2020	1.251657

Name: Total, dtype: float64

```
In [24]: range_ = (total_sales_by_year[2016] - total_sales_by_year[2020]) * 1_000_000_000
```

```
Out[24]: 307150222.0
```

```
In [25]: decline = (total_sales_by_year[2019] - total_sales_by_year[2020]) * 1_000_000_000
```

```
Out[25]: 207082929.0000003
```

```
In [26]: percentage_decline = round((decline / (1_000_000_000 * total_sales_by_year[2019])) * 100, 1)
```

```
Out[26]: 14.2
```

```
In [27]: growth = (total_sales_by_year[2021] - total_sales_by_year[2020]) * 1_000_000_000
```

```
Out[27]: 125366156.00000012
```

```
In [28]: percentage_growth = round((growth / (1_000_000_000 * total_sales_by_year[2020])) * 100, 1)
percentage_growth
```

```
Out[28]: 10.0
```

## Summary of the Sales Analysis

- For the period under review, the sales peaked in year 2016 reaching a tune of 1.55 billion USD
- The lowest point of the sales was in year 2020 with a total sales of 1.25 billion USD. This may be as a result of the impact of covid-19 to business.
- The sales declined from 1.46 billion USD in 2019 to 1.25 billion USD in 2020. Which means that the business suffered 14.2% decline from 2019 to 2020
- However the business grew by 10.0% from 2020 to 2021. Hence the total sales in 2021 was 1.38 billion USD

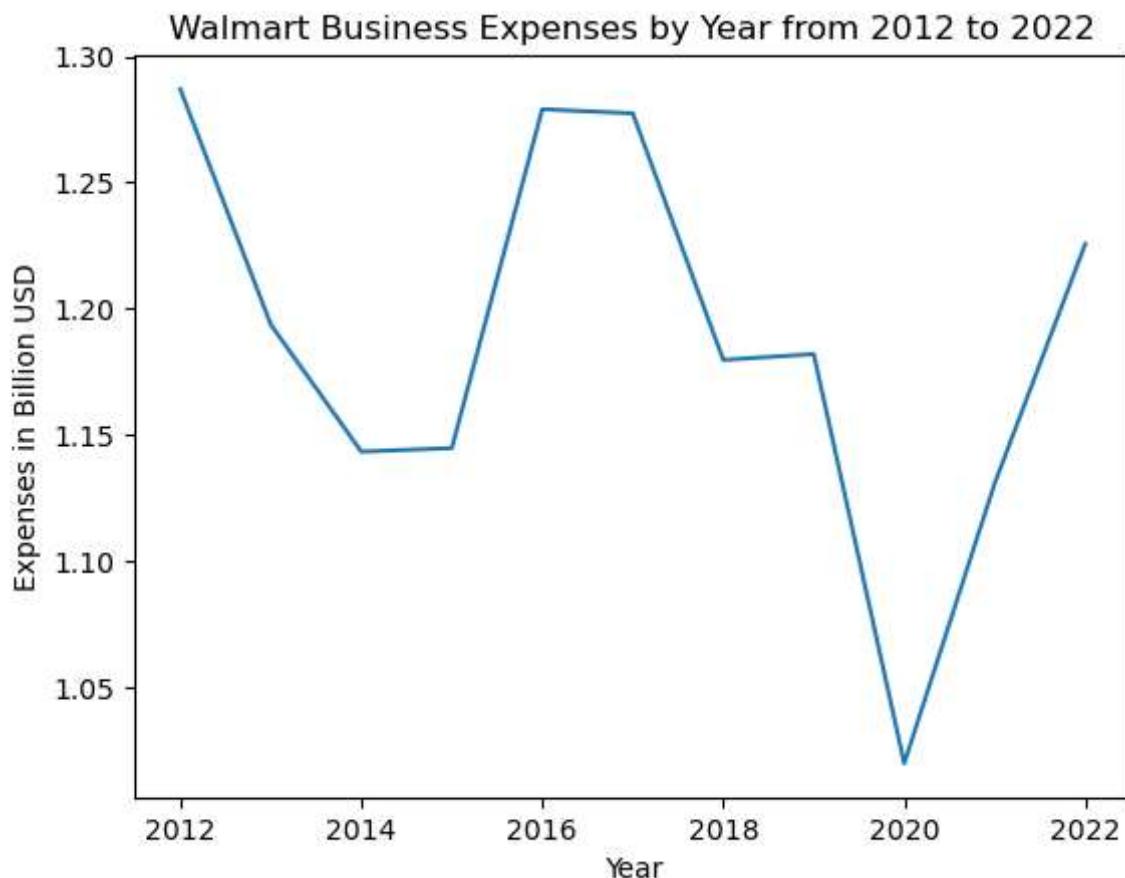
## Cost Analysis and Visualization

```
In [29]: cost = df[df['Account'] != 'Sales']
```

```
In [30]: cost_by_year = cost.groupby('Year')['Total'].sum() / 1_000_000_000
cost_by_year.sort_values(ascending=False)
```

```
Out[30]: Year
2012    1.286976
2016    1.279029
2017    1.277387
2022    1.225602
2013    1.193732
2019    1.182028
2018    1.179739
2015    1.144769
2014    1.143391
2021    1.130884
2020    1.019909
Name: Total, dtype: float64
```

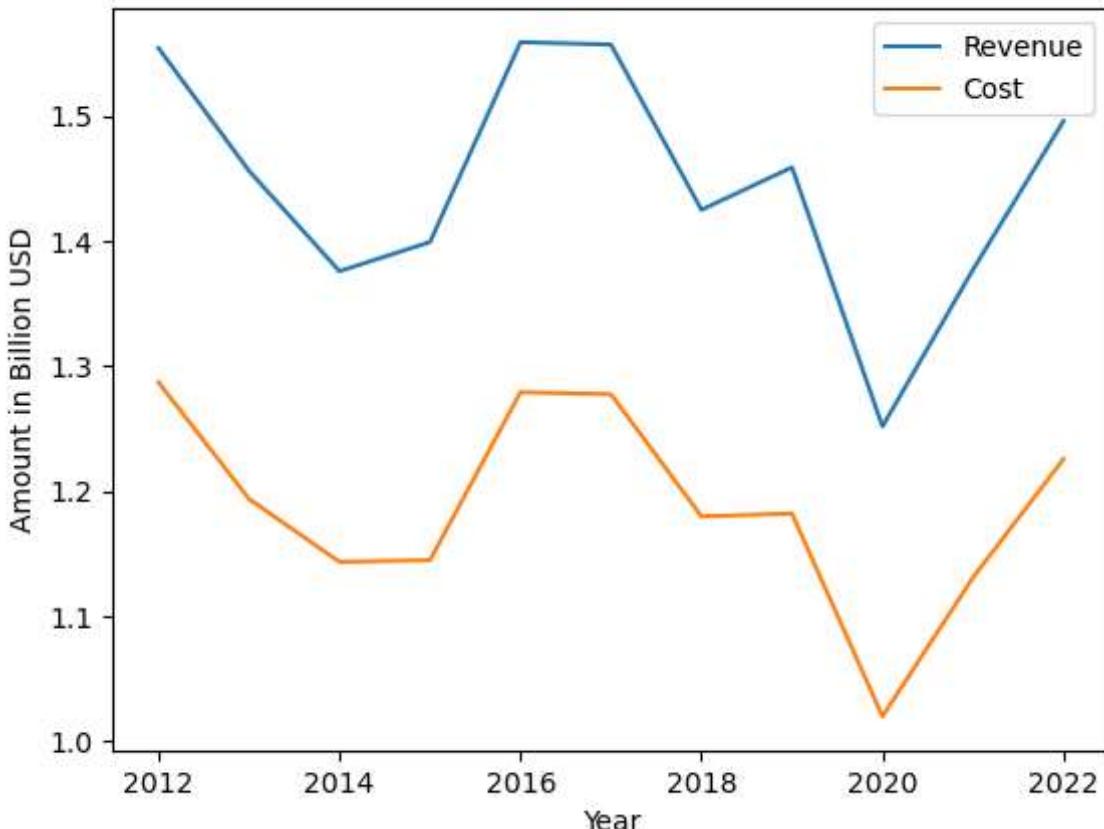
```
In [31]: cost_by_year.plot()
plt.title('Walmart Business Expenses by Year from 2012 to 2022')
plt.xlabel('Year')
plt.ylabel('Expenses in Billion USD')
plt.savefig('cost.png')
plt.show()
```



```
In [32]: total_sales_by_year.name = 'Revenue'  
cost_by_year.name = 'Cost'
```

```
In [33]: total_sales_by_year.plot()  
cost_by_year.plot()  
plt.title('Walmart Cost and Revenue from 2012 - 2022')  
plt.xlabel('Year')  
plt.ylabel('Amount in Billion USD')  
plt.legend()  
plt.savefig('sale_cost.png')  
plt.show()
```

### Walmart Cost and Revenue from 2012 - 2022



## Summary of the Cost Analysis

- The cost of running Walmart business peaked in 2012 with to about 1.29 billion USD.
- The business spent 1.02 billion USD to run the business in 2020. This was the least that was spent for the period under review. The spending maybe as a result of the impact of covid-19.
- The cost of running the business for the period under review was always lower than the revenue. This means that the business made considerable amount of profit each.

## Profit Analysis and Visualization

```
In [34]: profit = total_sales_by_year - cost_by_year
```

```
In [35]: profit.name = 'Profit'
```

```
In [36]: profit.sort_values(ascending=False) * 1000
```

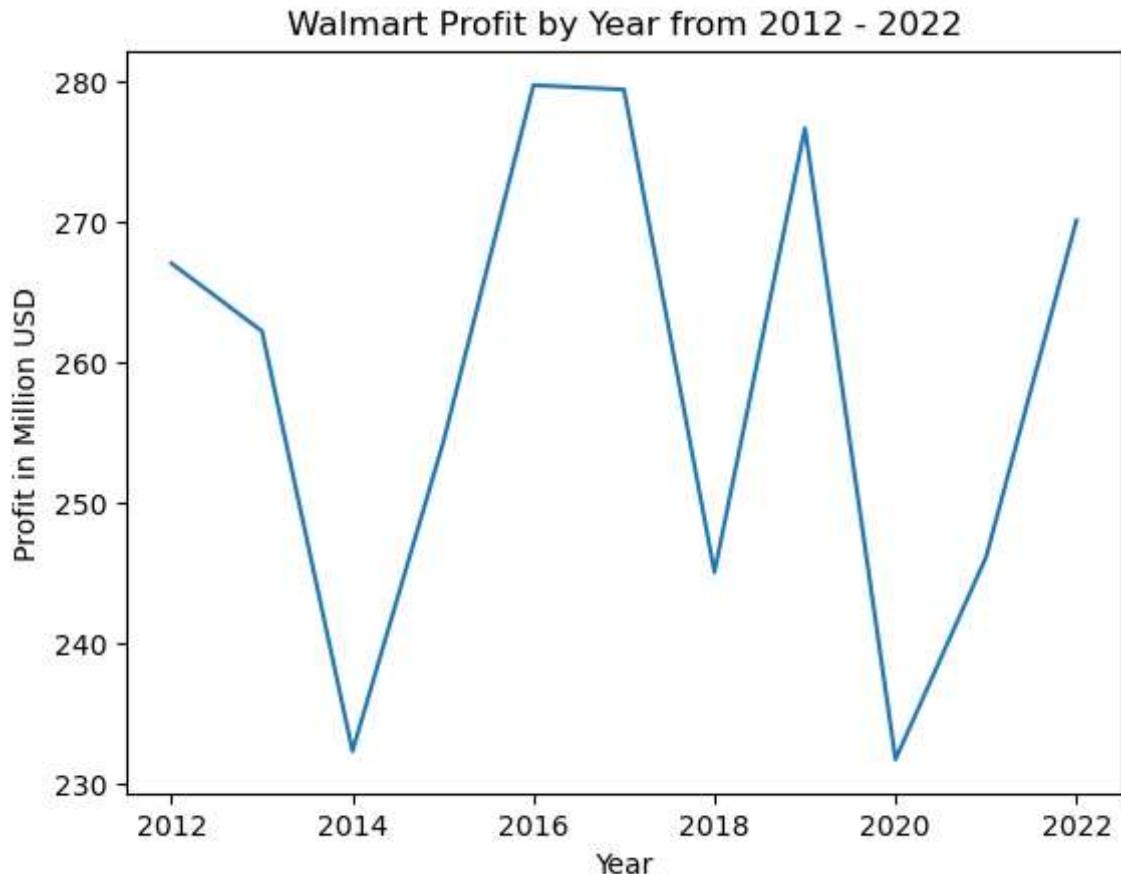
Out[36]:

Year	Profit (Billion USD)
2016	279.777704
2017	279.462000
2019	276.712132
2022	270.146778
2012	267.084281
2013	262.251784
2015	254.269986
2021	246.138623
2018	245.079725
2014	232.336858
2020	231.748250

Name: Profit, dtype: float64

In [37]:

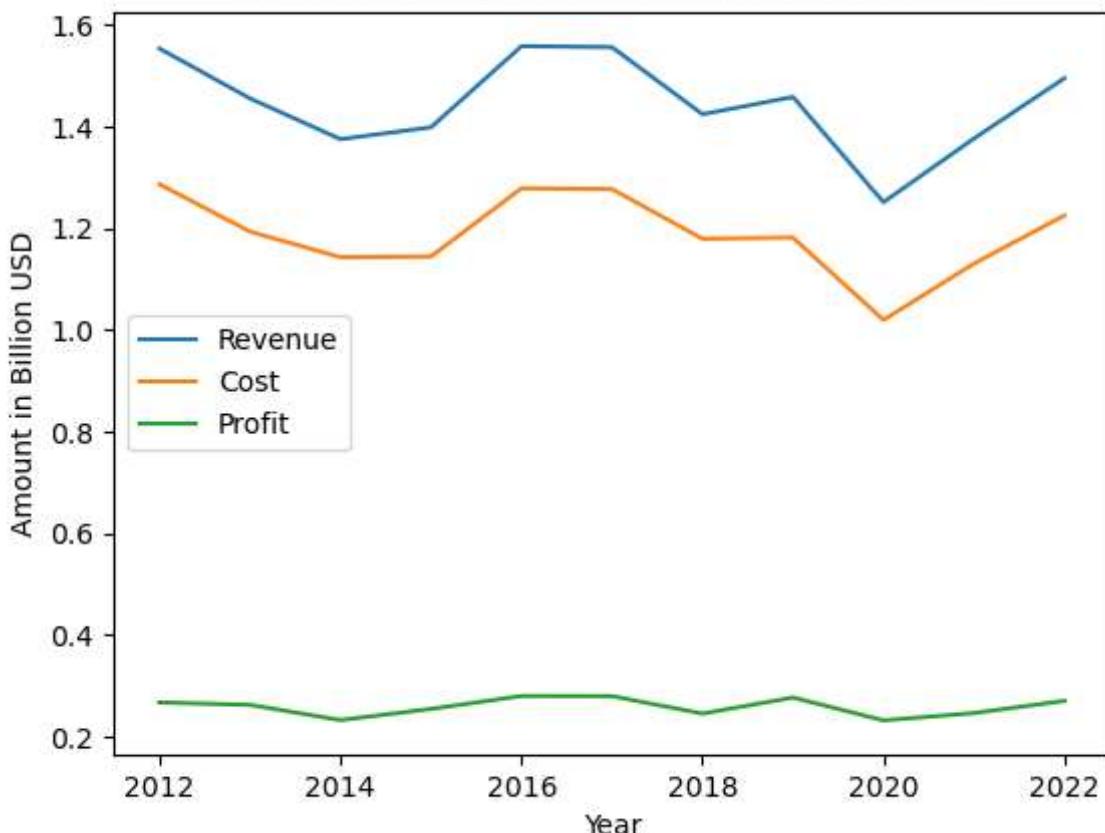
```
(profit * 1000).plot()  
plt.title('Walmart Profit by Year from 2012 - 2022')  
plt.xlabel('Year')  
plt.ylabel("Profit in Million USD")  
plt.savefig('profit.png')  
plt.show()
```



In [38]:

```
total_sales_by_year.plot()  
cost_by_year.plot()  
profit.plot()  
plt.title('Walmart Cost and Revenue from 2012 - 2022')  
plt.xlabel('Year')  
plt.ylabel('Amount in Billion USD')  
plt.legend()  
plt.savefig('sales_cost_profit.png')  
plt.show()
```

### Walmart Cost and Revenue from 2012 - 2022



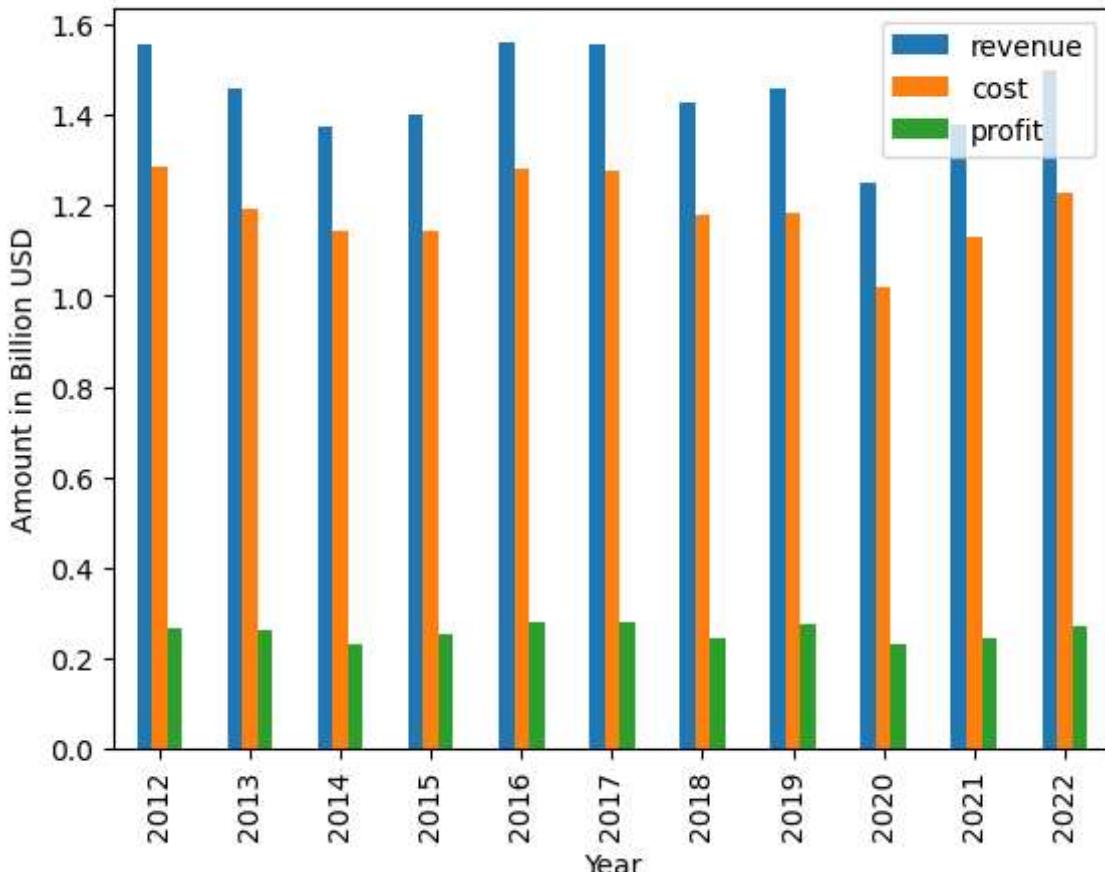
```
In [39]: summary = pd.DataFrame({'revenue': total_sales_by_year, 'cost': cost_by_year, 'profit': profit})
summary.head()
```

```
Out[39]:
```

Year	revenue	cost	profit
2012	1.554061	1.286976	0.267084
2013	1.455984	1.193732	0.262252
2014	1.375727	1.143391	0.232337
2015	1.399039	1.144769	0.254270
2016	1.558807	1.279029	0.279778

```
In [40]: summary.plot.bar()
plt.title('Walmart Cost and Revenue from 2012 - 2022')
plt.xlabel('Year')
plt.ylabel('Amount in Billion USD')
plt.savefig('sales_cost_profit2.png')
plt.show()
```

### Walmart Cost and Revenue from 2012 - 2022



```
In [41]: summary['percentage_profit'] = summary.profit / summary.cost * 100
summary
```

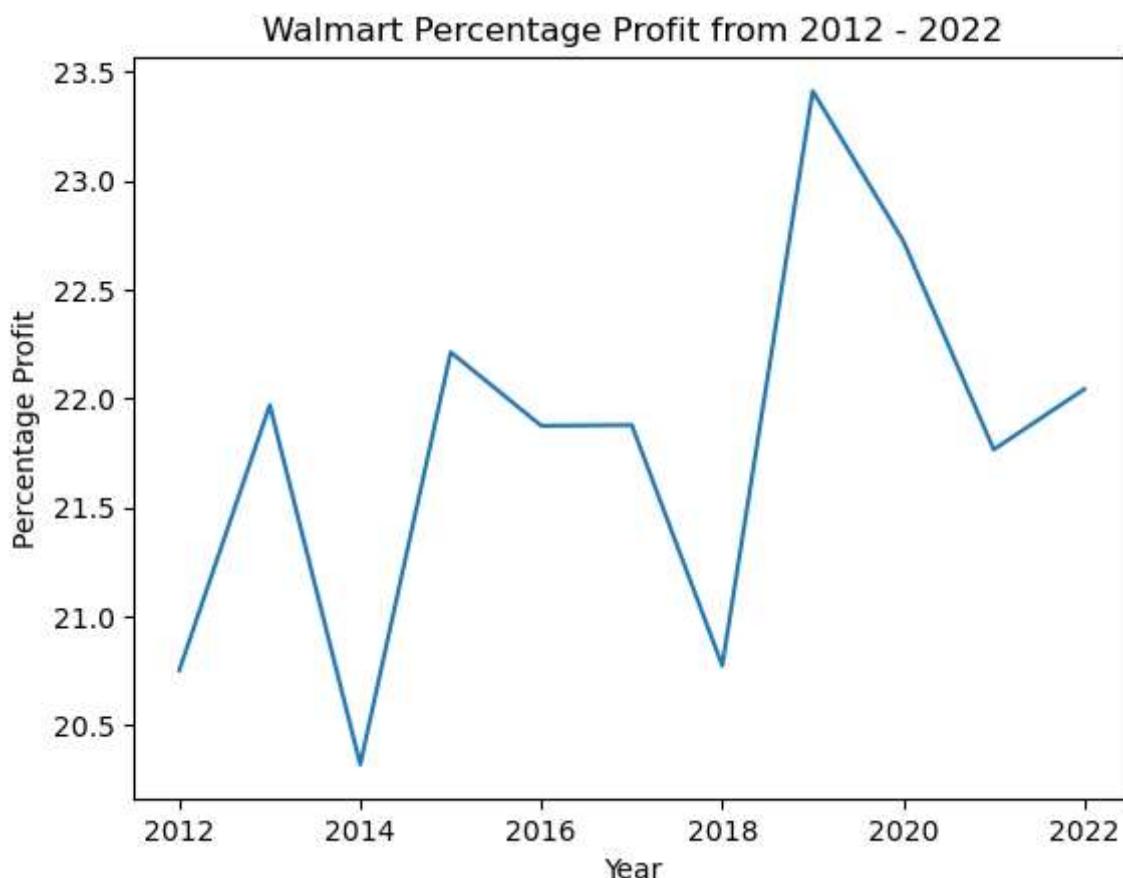
```
Out[41]:    revenue      cost      profit  percentage_profit
```

Year	revenue	cost	profit	percentage_profit
2012	1.554061	1.286976	0.267084	20.752850
2013	1.455984	1.193732	0.262252	21.969068
2014	1.375727	1.143391	0.232337	20.319991
2015	1.399039	1.144769	0.254270	22.211465
2016	1.558807	1.279029	0.279778	21.874222
2017	1.556849	1.277387	0.279462	21.877628
2018	1.424819	1.179739	0.245080	20.774066
2019	1.458740	1.182028	0.276712	23.409956
2020	1.251657	1.019909	0.231748	22.722454
2021	1.377023	1.130884	0.246139	21.765146
2022	1.495748	1.225602	0.270147	22.041973

```
In [42]: summary['percentage_profit'].sort_values(ascending=False)
```

```
Out[42]: Year
2019    23.409956
2020    22.722454
2015    22.211465
2022    22.041973
2013    21.969068
2017    21.877628
2016    21.874222
2021    21.765146
2018    20.774066
2012    20.752850
2014    20.319991
Name: percentage_profit, dtype: float64
```

```
In [43]: summary['percentage_profit'].plot()
plt.title('Walmart Percentage Profit from 2012 - 2022')
plt.xlabel('Year')
plt.ylabel('Percentage Profit')
plt.savefig('percentage_profit.png')
```



## Summary of Profit Analysis

- The business made the highest profit in 2016 and the least profit in 2020
- Profit in 2016 was 279.8 million USD, while the profit in 2020 was 231.7 million USD
- The percentage profit in 2019 was the highest while the percentage profit in 2014 was the least
- The percentage profit in 2019 was 23.4% while that of 2014 was 20.3%
- The year 2020 ranked second in percentage profit at about 22.7%. This shows that the business performed well in terms of resource management in 2020.

```
In [ ]:
```