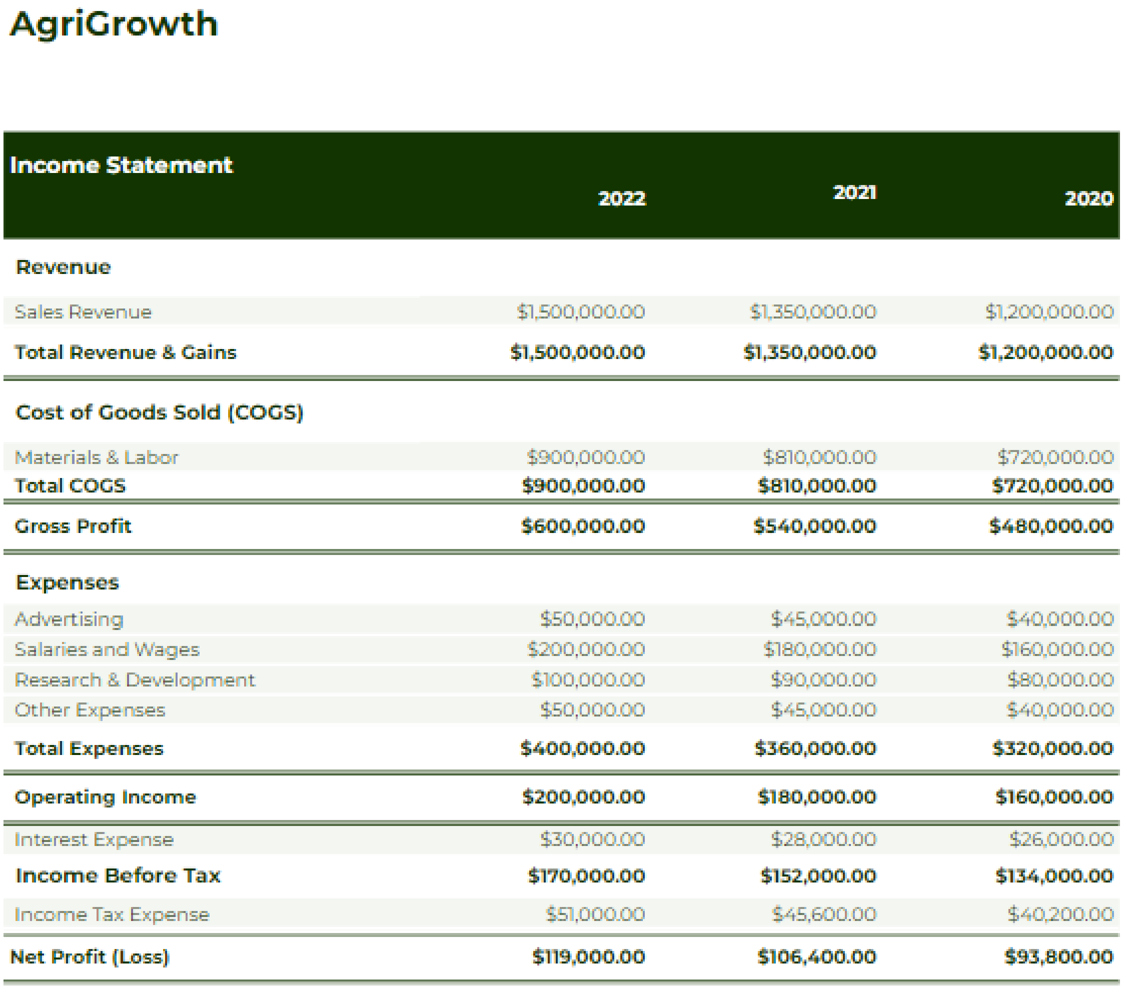
The image shows the data sources used to implement the tasks



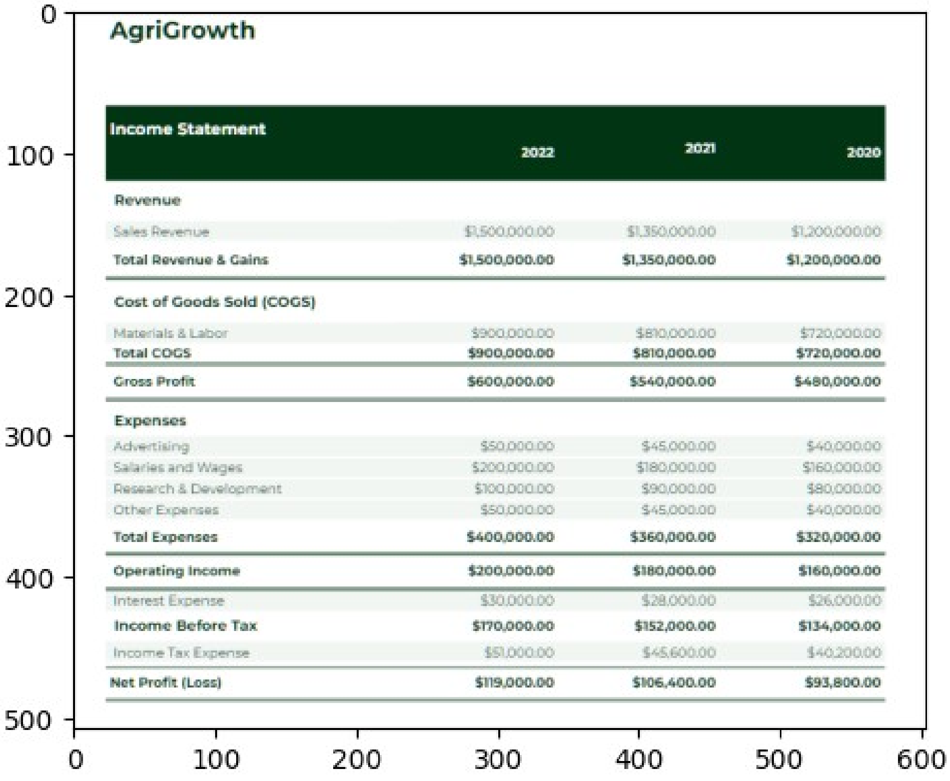
AgriGrowth

Income Statement	2022	2021	2020
Revenue			
Sales Revenue	$1,500,000.00	$1,350,000.00	$1,200,000.00
Total Revenue & Gains	**$1,500,000.00**	**$1,350,000.00**	**$1,200,000.00**
Cost of Goods Sold (COGS)			
Materials & Labor	$900,000.00	$810,000.00	$720,000.00
Total COGS	**$900,000.00**	**$810,000.00**	**$720,000.00**
Gross Profit	**$600,000.00**	**$540,000.00**	**$480,000.00**
Expenses			
Advertising	$50,000.00	$45,000.00	$40,000.00
Salaries and Wages	$200,000.00	$180,000.00	$160,000.00
Research & Development	$100,000.00	$90,000.00	$80,000.00
Other Expenses	$50,000.00	$45,000.00	$40,000.00
Total Expenses	**$400,000.00**	**$360,000.00**	**$320,000.00**
Operating Income	**$200,000.00**	**$180,000.00**	**$160,000.00**
Interest Expense	$30,000.00	$28,000.00	$26,000.00
Income Before Tax	**$170,000.00**	**$152,000.00**	**$134,000.00**
Income Tax Expense	$51,000.00	$45,600.00	$40,200.00
Net Profit (Loss)	**$119,000.00**	**$106,400.00**	**$93,800.00**

```
!pip install opencv-python
import cv2
import matplotlib.pyplot as plt

# Replace 'path/to/your/image.jpg' with the actual path to your image
image_path = '/content/Income Statements.PNG'

# Read the image using OpenCV
image = cv2.imread(image_path)

# Check if image is read successfully
if image is None:
    print("Error: Could not read image from", image_path)
```

```
else:
    # Display the image using matplotlib
    plt.imshow(image)
    plt.show()
```

```
Requirement already satisfied: opencv-python in
/usr/local/lib/python3.10/dist-packages (4.8.0.76)
Requirement already satisfied: numpy>=1.21.2 in
/usr/local/lib/python3.10/dist-packages (from opencv-python) (1.25.2)
```

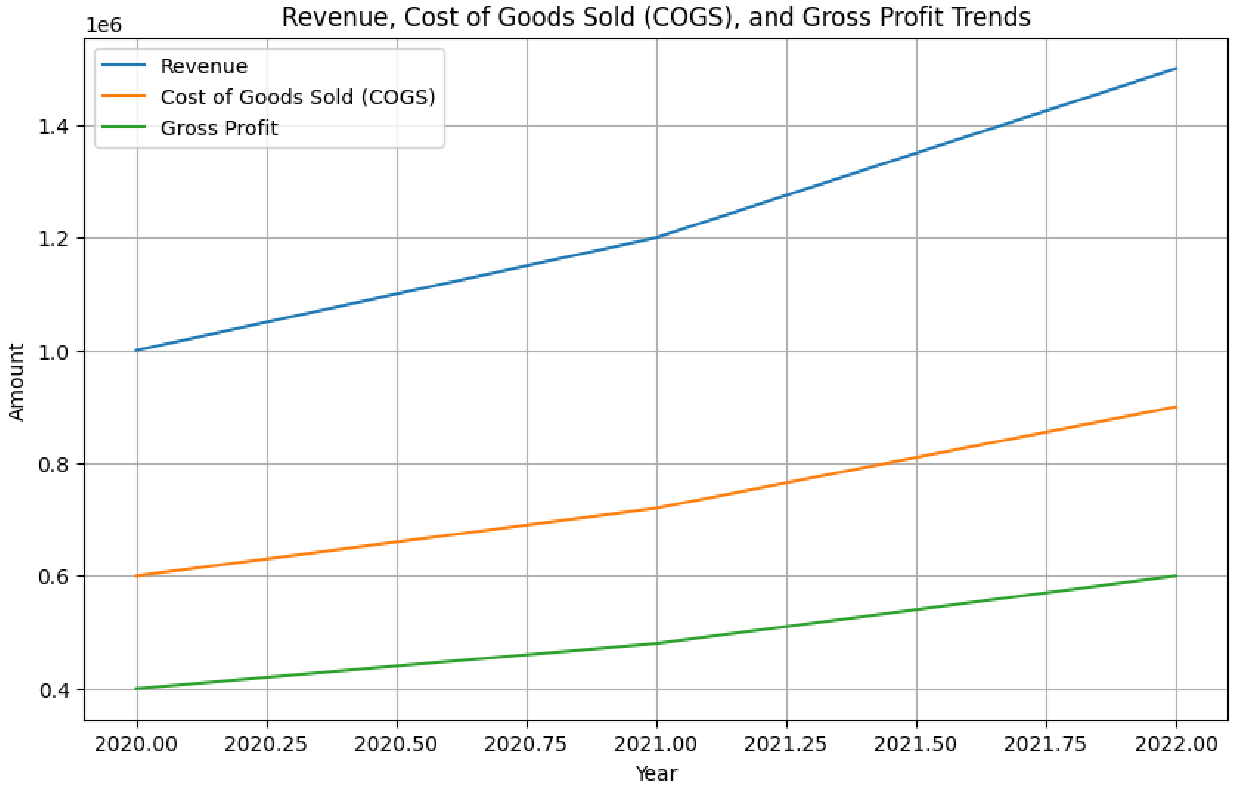

0
100
200
300
400
500
0
100
200
300
400
500
600

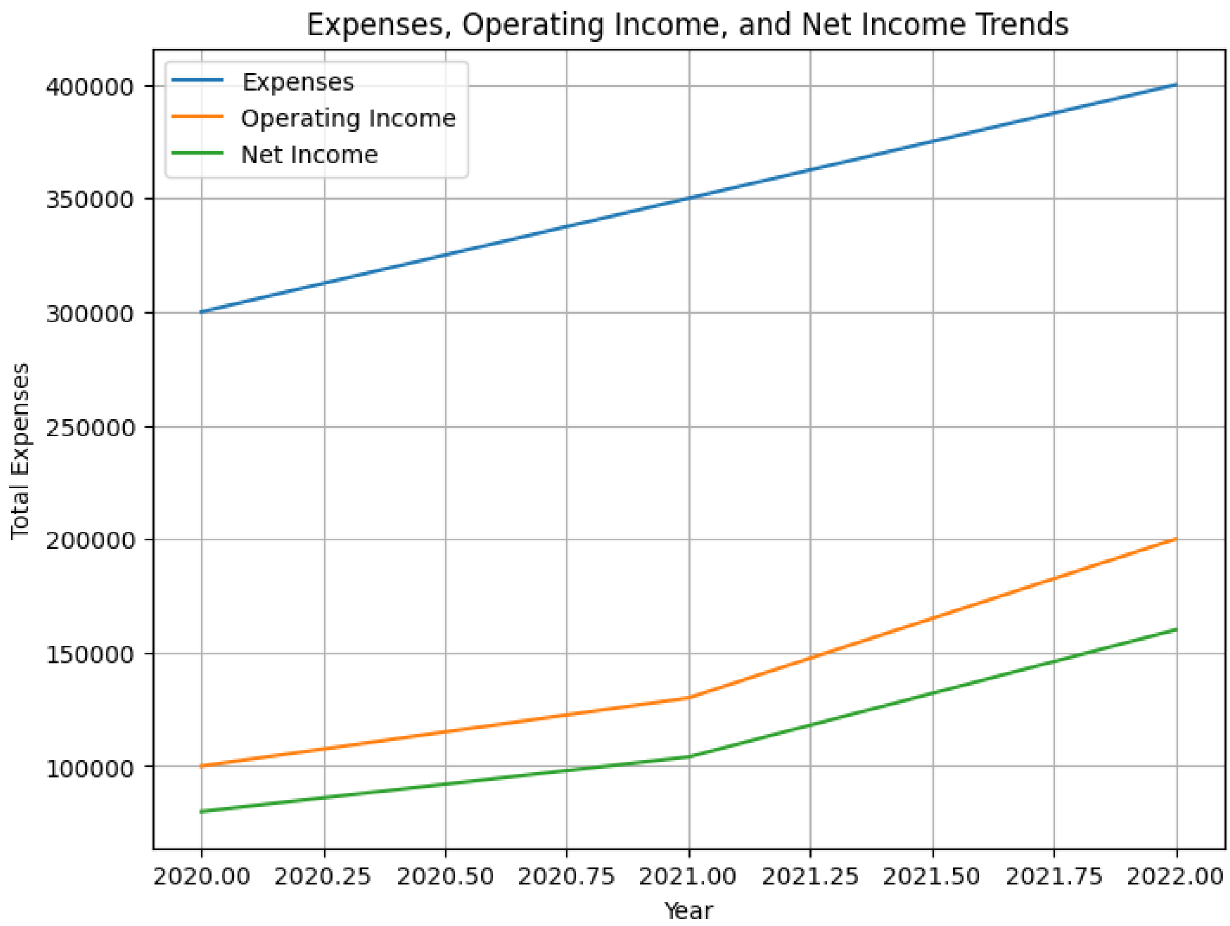
AgriGrowth

Income Statement	2022	2021	2020
Revenue			
Sales Revenue	$1,500,000.00	$1,350,000.00	$1,200,000.00
Total Revenue & Gains	**$1,500,000.00**	**$1,350,000.00**	**$1,200,000.00**
Cost of Goods Sold (COGS)			
Materials & Labor	$900,000.00	$810,000.00	$720,000.00
Total COGS	**$900,000.00**	**$810,000.00**	**$720,000.00**
Gross Profit	**$600,000.00**	**$540,000.00**	**$480,000.00**
Expenses			
Advertising	$50,000.00	$45,000.00	$40,000.00
Salaries and Wages	$200,000.00	$180,000.00	$160,000.00
Research & Development	$100,000.00	$90,000.00	$80,000.00
Other Expenses	$50,000.00	$45,000.00	$40,000.00
Total Expenses	**$400,000.00**	**$360,000.00**	**$320,000.00**
Operating Income	**$200,000.00**	**$180,000.00**	**$160,000.00**
Interest Expense	$30,000.00	$28,000.00	$26,000.00
Income Before Tax	**$170,000.00**	**$152,000.00**	**$134,000.00**
Income Tax Expense	$51,000.00	$45,600.00	$40,200.00
Net Profit (Loss)	**$119,000.00**	**$106,400.00**	**$93,800.00**

```
import matplotlib.pyplot as plt

# Financial data
# A code that implements Trends Over Time
years = [2020, 2021, 2022]
revenue = [1000000, 1200000, 1500000]
cogs = [600000, 720000, 900000]
gross_profit = [revenue[i] - cogs[i] for i in range(len(revenue))]
expenses = [300000, 350000, 400000]
operating_income = [gross_profit[i] - expenses[i] for i in
range(len(gross_profit))]
net_income = [operating_income[i] * 0.8 for i in
range(len(operating_income))]  # Assuming a tax rate of 20%

# Line chart for Revenue, Cost of Goods Sold (COGS), and Gross Profit
```

```
plt.figure(figsize=(10, 6))
plt.plot(years, revenue, label='Revenue')
plt.plot(years, cogs, label='Cost of Goods Sold (COGS)')
plt.plot(years, gross_profit, label='Gross Profit')
plt.xlabel('Year')
plt.ylabel('Amount')
plt.title('Revenue, Cost of Goods Sold (COGS), and Gross Profit
Trends')
plt.legend()
plt.grid(True)
plt.show()

# Line chart for Expenses, Operating Income, and Net Income
plt.figure(figsize=(8, 6))
plt.plot(years, expenses, label='Expenses')
plt.plot(years, operating_income, label='Operating Income')
plt.plot(years, net_income, label='Net Income')
plt.xlabel('Year')
plt.ylabel('Total Expenses')
plt.title('Expenses, Operating Income, and Net Income Trends')
plt.legend()
plt.grid(True)
plt.show()

# You can create similar charts for other financial metrics
```



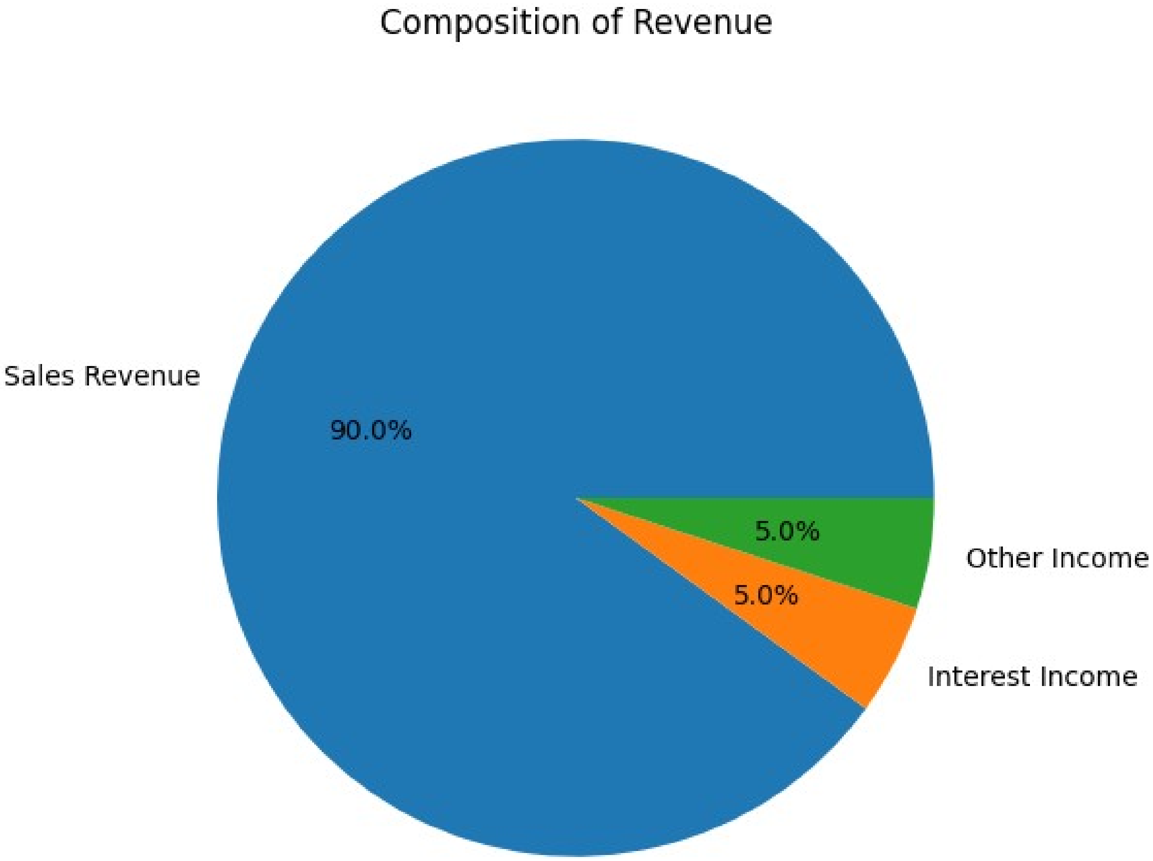


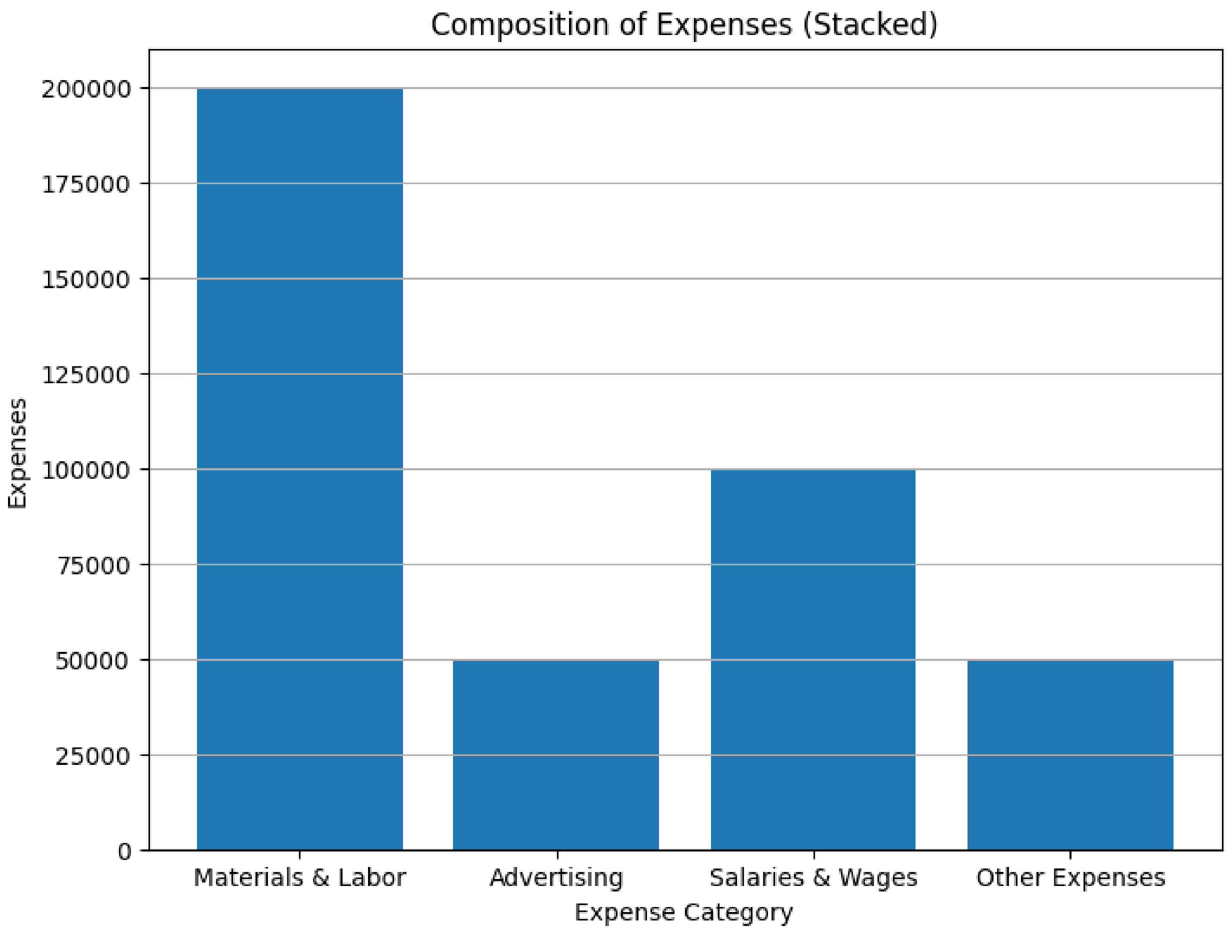
Trends Over Time

- The first chart shows the trends in Revenue, COGS, and Gross Profit over the three years.
- The second chart shows the trends in Expenses, Operating Income, and Net Income.

```
import matplotlib.pyplot as plt

# Financial data
# Composition of Revenue and Expenses
revenue_sources = ["Sales Revenue", "Interest Income", "Other Income"]
revenue_amounts = [900000, 50000, 50000]

expense_categories = ["Materials & Labor", "Advertising", "Salaries &
Wages", "Other Expenses"]
expense_amounts = [200000, 50000, 100000, 50000]

# Pie chart for Revenue Sources
plt.figure(figsize=(6, 6))
plt.pie(revenue_amounts, labels=revenue_sources, autopct="%1.1f%%")
plt.title('Composition of Revenue')
plt.show()
```

```
# Stacked bar chart for Expense Categories
plt.figure(figsize=(8, 6))
plt.bar(expense_categories, expense_amounts)
plt.xlabel('Expense Category')
plt.ylabel('Expenses')
plt.title('Composition of Expenses (Stacked)')
plt.grid(axis='y')
plt.show()
```



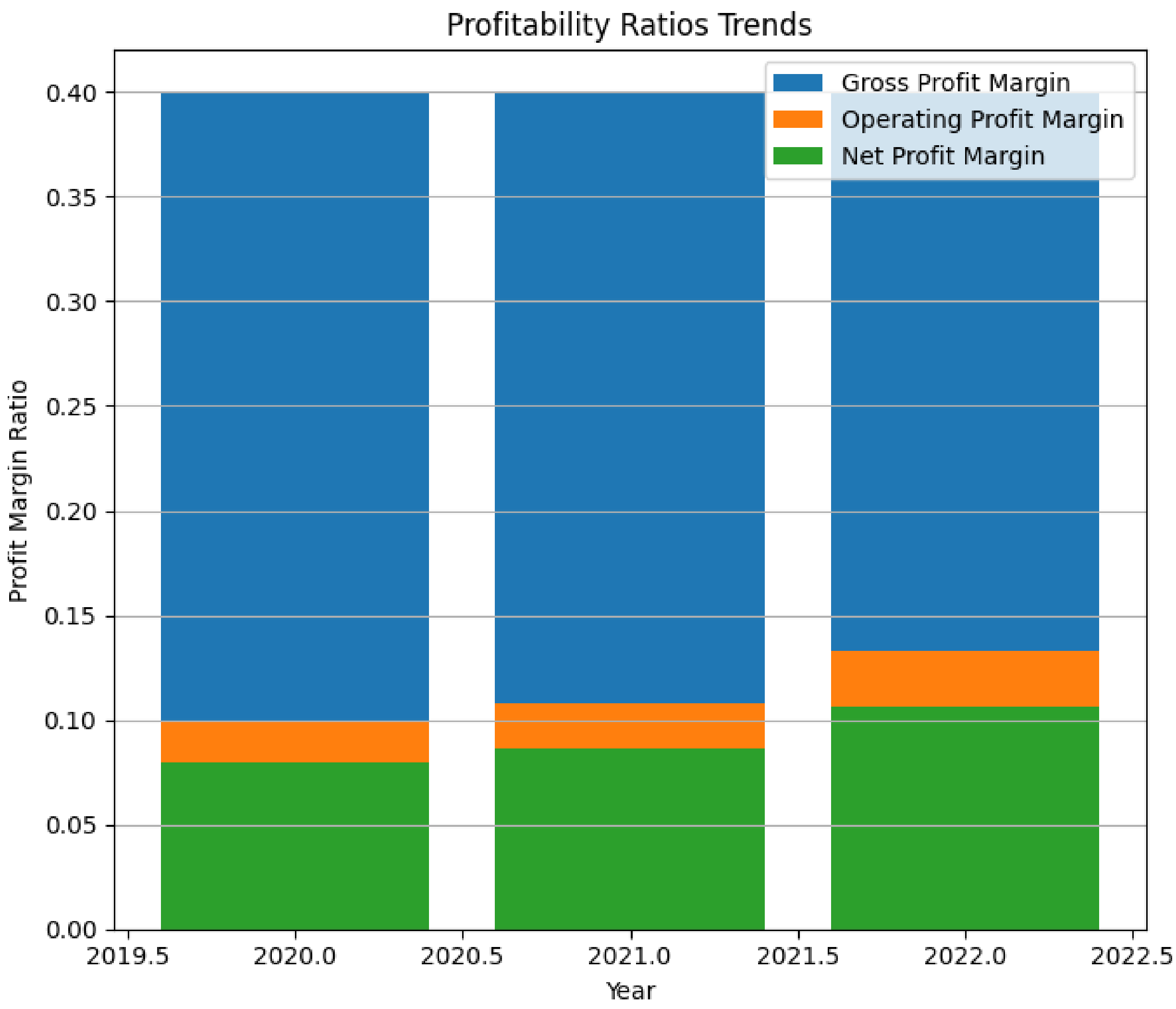


Composition of Revenue and Expenses

- A pie chart shows the composition of revenue from different sources. Each slice of the pie represents a revenue source, and the size of the slice corresponds to its percentage contribution to the total revenue.
- A stacked bar chart shows the composition of expenses by category. Each bar is divided into sections representing different expense categories, with the height of each section representing the amount spent on that category.

```
import matplotlib.pyplot as plt

# Financial data
# Profitability Ratios
years = [2020, 2021, 2022]
revenue = [1000000, 1200000, 1500000]
cogs = [600000, 720000, 900000]
expenses = [300000, 350000, 400000]

# Calculate profitability ratios
gross_profit_margin = [(revenue[i] - cogs[i]) / revenue[i] for i in
range(len(revenue))]
operating_profit_margin = [(gross_profit_margin[i] - (expenses[i] /
revenue[i])) for i in range(len(gross_profit_margin))]
net_profit_margin = [operating_profit_margin[i] * 0.8 for i in
range(len(operating_profit_margin))]  # Assuming a tax rate of 20%
rounded_gross_profit_margin = [round(value, 4) for value in
gross_profit_margin]
rounded_operating_profit_margin = [round(value, 4) for value in
operating_profit_margin]
rounded_net_profit_margin = [round(value, 4) for value in
net_profit_margin]
print('The gross_profit_margin is', rounded_gross_profit_margin)
print('The operating_profit_margin is',
rounded_operating_profit_margin)
print('The net_profit_margin is', rounded_net_profit_margin)


# Bar chart for Profitability Ratios
plt.figure(figsize=(7, 6))
plt.bar(years, gross_profit_margin, label='Gross Profit Margin')
plt.bar(years, operating_profit_margin, label='Operating Profit
Margin')
plt.bar(years, net_profit_margin, label='Net Profit Margin')
plt.xlabel('Year')
plt.ylabel('Profit Margin Ratio')
plt.title('Profitability Ratios Trends')
plt.legend()
plt.grid(axis='y')
plt.xticks(rotation=0)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```

```
The gross_profit_margin is [0.4, 0.4, 0.4]
The operating_profit_margin is [0.1, 0.1083, 0.1333]
The net_profit_margin is [0.08, 0.0867, 0.1067]
```



Profitability Ratios

The bar chart allows us to visualize trends in the company's profitability. A consistently high gross profit margin indicates efficient cost management in production. An increasing operating profit margin suggests the company is controlling its operating expenses effectively. A rising net profit margin reflects an improvement in the company's overall profitability.