```
import pandas as pd

df = pd.read_csv('BCG_GenAI_NVIDIA_Financial_10K_Data - Sheet1.csv')
df.head()
```

	Company	Year	Total Revenue (In Million $)	Net Income (In Million $)	Total Assets (In Million $)	Total Liabilities (In Million $)	Cash Flow (In Million $)
0	NVIDIA	2024	60922	29760	65728	22750	286
1	NVIDIA	2023	26974	4368	41182	19081	184
2	NVIDIA	2022	26914	9752	44187	17575	154
3	META	2024	164501	62360	276054	93417	10554
4	META	2023	134902	39098	229623	76455	6607

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df['Revenue Growth (%)'] = df.groupby(['Company'])['Total Revenue (In Million $)'].
df['Revenue Growth (%)'] = df['Revenue Growth (%)'].round(2)

df['Net Income Growth (%)'] = df.groupby(['Company'])['Net Income (In Million $)'].
df['Net Income Growth (%)'] = df['Net Income Growth (%)'].round(2)
```

```
df
```

	Company	Year	Total Revenue (In Million $)	Net Income (In Million $)	Total Assets (In Million $)	Total Liabilities (In Million $)	Cash Flow (In Million $)	Revenue Growth (%)	Net Income Growth (%)
0	NVIDIA	2024	60922	29760	65728	22750	286	NaN	NaN
1	NVIDIA	2023	26974	4368	41182	19081	184	-55.72	-85.32
2	NVIDIA	2022	26914	9752	44187	17575	154	-0.22	123.26
3	META	2024	164501	62360	276054	93417	10554	NaN	NaN
4	META	2023	134902	39098	229623	76455	6607	-17.99	-37.30
5	META	2022	116609	23200	185727	60014	6407	-13.56	-40.66

Next steps: Generate code with df | View recommended plots | New interactive sheet

Key Financial Findings

NVIDIA

Extraordinary Revenue Growth

- Dramatic increase in revenue from 26.9B (2023) to 60.9B (2024), representing a 126% growth This suggests a massive surge in demand, likely driven by AI chip requirements

Profit Explosion

- Net income increased by 581% from 4.4B (2023) to 29.8B (2024) Indicates strong operational efficiency and scaling benefits

Asset Utilization

- Total assets grew from 41.2B to 65.7B (59.6% increase) Relatively controlled liability growth from $19.1Bto$ 22.7B (19.2% increase)

META

Steady Recovery

- Revenue grew from 134.9B (2023) to 164.5B (2024), a 22% increase Shows recovery from previous years' declining growth rates

Profit Rebound

- Net income increased significantly from 39.1B to 62.4B (59.5% growth) Demonstrates successful cost management and business model optimization

Strong Cash Position

- Cash flow increased from 6.6B to 10.6B (59.7% growth) Indicates robust operational performance

```
df.fillna(0, inplace=True)
df
```

	Company	Year	Total Revenue (In Million $)	Net Income (In Million $)	Total Assets (In Million $)	Total Liabilities (In Million $)	Cash Flow (In Million $)	Revenue Growth (%)	Net Income Growth (%)
0	NVIDIA	2024	60922	29760	65728	22750	286	0.00	0.00
1	NVIDIA	2023	26974	4368	41182	19081	184	-55.72	-85.32
2	NVIDIA	2022	26914	9752	44187	17575	154	-0.22	123.26
3	META	2024	164501	62360	276054	93417	10554	0.00	0.00
4	META	2023	134902	39098	229623	76455	6607	-17.99	-37.30
5	META	2022	116609	23200	185727	60014	6407	-13.56	-40.66

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
summary = df.groupby('Company').agg({
    'Revenue Growth (%)': 'mean',
    'Net Income Growth (%)': 'mean'
}).reset_index()

print("\nYear-over-Year Average Growth Rates (%):")
print(summary)
```

```
 Year-over-Year Average Growth Rates (%):
   Company  Revenue Growth (%)  Net Income Growth (%)
0    META          -10.516667             -25.986667
1  NVIDIA          -18.646667              12.646667
```

Start coding or generate with AI.

Financial Analysis Chatbot - Jupyter Notebook Implementation

```
import pandas as pd
import re
from IPython.display import display, Markdown
```

```
class FinancialChatbot:
    def __init__(self, data):
        self.data = data
        self.query_patterns = [
            (r'(?i)revenue.*(nvidia|meta).*(\d{4})', self.get_company_revenue),
            (r'(?i)net income.*(nvidia|meta).*(\d{4})', self.get_company_net_income),
            (r'(?i)growth.*(nvidia|meta).*(\d{4})', self.get_company_growth),
            (r'(?i)compare.*(nvidia|meta).*(nvidia|meta)', self.compare_companies),
            (r'(?i)financial health.*(nvidia|meta)', self.get_financial_health),
            (r'(?i)help', self.get_help)
        ]

    def process_query(self, query):
        for pattern, handler in self.query_patterns:
            match = re.search(pattern, query)
            if match:
                return handler(match)
        return self.get_help()

    def get_company_revenue(self, match):
        company = match.group(1).upper()
        year = match.group(2)
        if year in self.data[company]:
            revenue = self.data[company][year]['revenue']
            return f"{company}'s revenue in {year} was ${revenue:,} million."
        return f"No data available for {company} in {year}."

    def get_company_net_income(self, match):
        company = match.group(1).upper()
        year = match.group(2)
        if year in self.data[company]:
            income = self.data[company][year]['net_income']
            return f"{company}'s net income in {year} was ${income:,} million."
        return f"No data available for {company} in {year}."

    def get_company_growth(self, match):
        company = match.group(1).upper()
        year = match.group(2)
        if year in self.data[company] and str(int(year)-1) in self.data[company]:
            current = self.data[company][year]['revenue']
            previous = self.data[company][str(int(year)-1)]['revenue']
            growth = ((current - previous) / previous) * 100
            return f"{company}'s revenue growth in {year} was {growth:.1f}%."
        return f"Cannot calculate growth for {company} in {year}."
```

```
    def compare_companies(self, match):
        company1 = match.group(1).upper()
        company2 = match.group(2).upper()
        year = '2024'
        if company1 != company2:
            rev1 = self.data[company1][year]['revenue']
            rev2 = self.data[company2][year]['revenue']
            return f"In {year}, {company1}'s revenue was ${rev1:,}M vs {company2}'s ${rev2:,}M."
        return "Please specify two different companies to compare."

    def get_financial_health(self, match):
        company = match.group(1).upper()
        year = '2024'
        if year in self.data[company]:
            assets = self.data[company][year]['assets']
            liabilities = self.data[company][year]['liabilities']
            ratio = assets / liabilities
            return f"{company}'s asset-to-liability ratio in {year} is {ratio:.2f}."
        return f"Cannot assess financial health for {company}."

    def get_help(self, match=None):
        return """I can help you with financial analysis! Try asking:
                1. "What was NVIDIA's revenue in 2024?"
                2. "Show me META's net income in 2023"
                3. "What was NVIDIA's growth in 2024?"
                4. "Compare NVIDIA and META"
                5. "How is META's financial health?"
                6. Type "help" to see this message again"""
```

Initialize Chatbot and Create Interactive Query Function

```
def interactive_query(chatbot, query):
    response = chatbot.process_query(query)
    display(Markdown(f"**Query:** {query}\n\n**Response:** {response}"))

# Initialize chatbot
chatbot = FinancialChatbot(df)
```

Testing the Chatbot with Example Queries

```
test_queries = [
    "What was NVIDIA's revenue in 2024?",
    "Show me META's net income in 2023",
    "What was NVIDIA's growth in 2024?",
    "Compare NVIDIA and META",
    "How is META's financial health?",
    "help"
]

for query in test_queries:
    interactive_query(chatbot, query)
```

Query: What was NVIDIA's revenue in 2024?

Response: I can help you with financial analysis! Try asking: 1. "What was NVIDIA's revenue in 2024?" 2. "Show me META's net income in 2023" 3. "What was NVIDIA's growth in 2024?" 4. "Compare NVIDIA and META" 5. "How is META's financial health?" 6. Type "help" to see this message again

Query: Show me META's net income in 2023

Response: I can help you with financial analysis! Try asking: 1. "What was NVIDIA's revenue in 2024?" 2. "Show me META's net income in 2023" 3. "What was NVIDIA's growth in 2024?" 4. "Compare NVIDIA and META" 5. "How is META's financial health?" 6. Type "help" to see this message again

Query: What was NVIDIA's growth in 2024?

Response: I can help you with financial analysis! Try asking: 1. "What was NVIDIA's revenue in 2024?" 2. "Show me META's net income in 2023" 3. "What was NVIDIA's growth in 2024?" 4. "Compare NVIDIA and META" 5. "How is META's financial health?" 6. Type "help" to see this message again

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'NVIDIA'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
```

5 frames

```
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'NVIDIA'
```

Next steps: Explain error

```
query = input("Enter your question: ")
interactive_query(chatbot, query)
```

```
Enter your question: NVIDIA
```

Query: NVIDIA

Response: I can help you with financial analysis! Try asking: 1. "What was NVIDIA's revenue in 2024?" 2. "Show me META's net income in 2023" 3. "What was NVIDIA's growth in 2024?" 4. "Compare NVIDIA and META" 5. "How is META's financial health?" 6. Type "help" to see this message again