# RESEARCH OF HOW PYTHON CAN BE USED TO ANALYSE DATA FROM MONGO DB

Python can be a powerful tool for analyzing data from MongoDB and a CSV file. To develop a program to analyze sales data, you would typically use Python libraries such as pandas, pymongo, and matplotlib or seaborn for visualization. Here's a basic guide on how you can structure this software.

**Steps to Develop the Software**

1. **Set up MongoDB Connection**
   * Use pymongo to connect to the MongoDB database.
   * Fetch the sales data from MongoDB.
   * If additional data is in a CSV file (e.g., product details), load it using pandas.
2. **Data Analysis Tasks**
   * Using the combined data, perform analysis to answer the questions:
     + **Most Sold Products**: Count sales of each product to determine the most sold products.
     + **Most Used Payment Method**: Count occurrences of each payment method to find the most commonly used one.
     + **Most Profitable Product Category**: Calculate the profit for each category to determine the most profitable one.
3. **Visualization (Optional)**
   * Use matplotlib or seaborn to visualize the results, which can help in understanding trends.

## Sample Code Outline

* Here’s a Python script outline to get started with this project:

### Import Required Libraries

*import pandas as pd*

*from pymongo import MongoClient*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

### 2. Set up MongoDB Connection and Load Data

# Connect to MongoDB

client = MongoClient('mongodb://localhost:27017/')

db = client['your\_database\_name']

sales\_collection = db['sales']

# Load sales data from MongoDB

sales\_data = pd.DataFrame(list(sales\_collection.find()))

# Load additional product data from CSV if needed

product\_data = pd.read\_csv('path\_to\_product\_data.csv')

# Merge data if necessary

# sales\_data = pd.merge(sales\_data, product\_data, on='product\_id', how='left')

### 3. Analysis

#### a. Find the Most Sold Products

*most\_sold\_products = sales\_data['product\_id'].value\_counts()*

*print("Most Sold Products:\n", most\_sold\_products.head())*

#### b. Determine the Most Used Payment Method

payment\_methods = sales\_data['payment\_method'].value\_counts()

print("Most Used Payment Method:\n", payment\_methods.head())

#### c. Find the Most Profitable Product Category

*Assuming sales\_data has price and cost columns and product\_data has category:*

*# Calculate profit for each sale*

*sales\_data['profit'] = sales\_data['price'] - sales\_data['cost']*

*# Merge category information if necessary*

*sales\_data = pd.merge(sales\_data, product\_data[['product\_id', 'category']], on='product\_id', how='left')*

*# Calculate total profit by category*

*profit\_by\_category = sales\_data.groupby('category')['profit'].sum()*

*print("Most Profitable Product Category:\n", profit\_by\_category.sort\_values(ascending=False).head())*

### Visualization (Optional)

*# Plot Most Sold Products*

*plt.figure(figsize=(10, 6))*

*most\_sold\_products.head(10).plot(kind='bar')*

*plt.title("Top 10 Most Sold Products")*

*plt.xlabel("Product ID")*

*plt.ylabel("Number of Sales")*

*plt.show()*

*# Plot Most Used Payment Method*

*plt.figure(figsize=(8, 5))*

*payment\_methods.plot(kind='bar')*

*plt.title("Most Used Payment Method")*

*plt.xlabel("Payment Method")*

*plt.ylabel("Frequency")*

*plt.show()*

*# Plot Profit by Category*

*plt.figure(figsize=(10, 6))*

*profit\_by\_category.sort\_values(ascending=False).head(10).plot(kind='bar')*

*plt.title("Top 10 Most Profitable Product Categories")*

*plt.xlabel("Category")*

*plt.ylabel("Total Profit")*

*plt.show()*

## Summary

This script will connect to MongoDB, pull data into a pandas DataFrame, and perform the required analyses. You can adjust the MongoDB connection settings and file paths as needed.

*For additional information on APA Style formatting, please consult the* [*APA Style Manual, 7th*](https://apastyle.apa.org/style-grammar-guidelines) *Edition.*