# Product Sales Analysis Tool

## Overview

This Product Sales Analysis Tool is a lightweight, modular Python application designed to analyze sales data from CSV files and generate comprehensive reports. The tool processes transaction records, calculates key performance metrics, and provides actionable business insights without requiring any external dependencies.

## Features

* **CSV Data Processing**: Reads and validates sales data from standard CSV files
* **Product Performance Analysis**: Sorts and ranks products by revenue
* **Category Analysis**: Groups products by category and calculates category-level metrics
* **Business Insights**: Generates actionable insights including top performers and underperformers
* **Detailed Reporting**: Creates comprehensive text reports with sales metrics
* **Console Summary**: Displays key findings directly in the console
* **Batch Processing**: Includes a Windows batch file for easy execution

## Requirements

* Python 3.6 or higher
* No external Python packages required

## File Structure

* simple\_main.py - Main orchestration module
* simple\_data\_reader.py - CSV file reading and data validation
* simple\_analyzer.py - Data analysis and insight generation
* simple\_report\_writer.py - Report generation and output formatting
* run\_sales\_analyzer.bat - Windows batch file for easy execution

## Installation

1. Clone or download all files to a directory of your choice
2. Ensure Python 3.6+ is installed and available in your PATH

No additional installation steps are required as the application uses only Python standard library modules.

## Usage

### Using the Batch File (Windows)

1. Double-click run\_sales\_analyzer.bat
2. When prompted, enter the name of your CSV file or press Enter to use the default (sales\_data.csv)
3. The analysis will run and generate a report named sales\_report.txt
4. Choose whether to view the report immediately when prompted

You can also specify the input file directly:

run\_sales\_analyzer.bat your\_sales\_data.csv

### Using Python Directly

python simple\_main.py <input\_file> <output\_file>

Example:

python simple\_main.py sales\_data.csv sales\_report.txt

## Input File Format

The CSV file should contain the following columns: - product\_id - Unique identifier for each product - product\_name - Name of the product - category - Product category - price - Unit price (numeric) - quantity - Number of units sold (numeric)

Example:

product\_id,product\_name,category,price,quantity  
P001,Wireless Mouse,Electronics,29.99,150  
P002,Office Chair,Furniture,199.99,45  
P003,USB-C Cable,Accessories,12.99,300

## Output

The tool generates a comprehensive text report containing:

1. **Summary Statistics**: Total products, categories, and revenue
2. **Category Breakdown**: Performance metrics for each product category
3. **Top Products**: Detailed information on the top-performing products
4. **Business Insights**: Actionable insights including:
   * Revenue contribution of top 20% of products
   * Average price point and quantity sold
   * Underperforming products that may need attention

## Error Handling

The application includes robust error handling for common issues: - Missing or inaccessible input files - Incorrectly formatted CSV data - Missing required columns - Invalid numeric values

## Extending the Tool

The modular design makes it easy to extend functionality:

* Add new analysis metrics in simple\_analyzer.py
* Support additional file formats by creating new reader modules
* Implement different report formats (HTML, JSON, etc.) by adding new writer modules

## 