Faysal Aziz

Python for Smart Reporting and Data Visualization

Build Reports Faster with Python, Pandas and AI-Assisted Coding in 2 days

FA-0389 · 2 days

Agentic & Generative AI · Software Development · Data & Analytics · Automation, RPA & Power Platform

---

Introduction

Modern reporting is no longer limited to spreadsheets filled with formulas and repetitive manual work. Organizations increasingly expect teams to clean data, generate insights, automate reporting workflows, and present information visually in ways that support faster decision-making. Python has become one of the most practical tools for this purpose because it allows semi-technical professionals to automate routine reporting tasks without needing to become full-time software developers.

This course is designed specifically for professionals who work with reports, spreadsheets, dashboards, and operational data but may not come from a programming background. The focus is practical, approachable, and highly hands-on. Participants will learn how to use Python for reporting automation, data cleanup, chart generation, and Excel report production using widely adopted libraries such as Pandas, Matplotlib, Seaborn, and OpenPyXL. The course also introduces the modern workflow of using AI chatbots such as OpenAI ChatGPT to help generate, explain, troubleshoot, and improve Python code safely and efficiently.

The training emphasizes real-world reporting scenarios rather than academic programming theory. Participants will learn how to work smarter by combining foundational Python skills with AI-assisted development techniques that significantly reduce the learning curve and increase productivity. Industry-relevant examples and practical reporting exercises will be used throughout the course. The instructor brings over 30 years of industry experience and focuses on practical, operationally relevant workflows that are actively used in modern reporting and analytics environments.

Recent industry trends continue to reinforce the importance of Python-based reporting automation, particularly around Pandas for data manipulation, Matplotlib and Seaborn for visualization, and OpenPyXL for Excel automation workflows.

Learning outcomes

By the end of this course, participants will be able to:

- Understand the role of Python in reporting and business automation
- Use basic Python syntax and logic for reporting tasks
- Work with datasets using Pandas DataFrames
- Import and export CSV and Excel files
- Clean and transform reporting data efficiently
- Filter, sort, group, and summarize business data
- Generate charts and visual reports using Matplotlib and Seaborn
- Automate Excel report formatting using OpenPyXL
- Create reusable reporting scripts for repetitive tasks
- Use AI chatbots such as ChatGPT to generate and troubleshoot Python code
- Write effective prompts for AI-assisted coding workflows
- Interpret and modify AI-generated Python scripts safely
- Build end-to-end automated reporting workflows
- Improve reporting productivity while minimizing manual spreadsheet work

Prerequisites

- Basic computer literacy
- Familiarity with Microsoft Excel
- Basic understanding of tables, rows, columns, and spreadsheets
- No prior programming experience required
- No prior data science or software development background required
- Participants should have a laptop with Python and Visual Studio Code installed
- Curiosity and willingness to experiment with data and automation tools

Training outline

1. Introduction to Python for Reporting Automation

Understanding Modern Reporting Challenges

1. Manual reporting limitations
2. Repetitive spreadsheet workflows

3. Reporting bottlenecks and operational inefficiencies
4. Introduction to reporting automation concepts

Why Python for Reporting

1. Python in business operations
2. Reporting and analytics use cases
3. Python versus traditional spreadsheet workflows
4. Real-world reporting automation examples
5. Current industry adoption trends

Understanding the Python Ecosystem

1. Python interpreter fundamentals
2. Jupyter Notebook overview
3. Visual Studio Code overview
4. Python package ecosystem
5. Introduction to reporting libraries

Using AI Chatbots for Coding Assistance

1. Introduction to ChatGPT-assisted development
2. AI-assisted scripting workflows
3. Writing effective prompts for Python generation
4. Understanding AI-generated code safely
5. Improving productivity with AI coding assistants
6. Limitations and risks of AI-generated code
7. Human validation and verification practices

2. Python Fundamentals for Semi-Technical Users

Python Syntax Basics

1. Variables and data types
2. Strings and numbers
3. Lists and dictionaries
4. Tuples and sets

5. Comments and readability

Basic Program Flow

1. Conditional statements
2. Loops and iterations
3. Basic logic building
4. Working with repetitive tasks

Functions and Reusability

1. Creating reusable functions
2. Function inputs and outputs
3. Organizing reporting scripts
4. Modular scripting concepts

Error Handling Fundamentals

1. Common Python errors
2. Reading error messages
3. Debugging basics
4. Troubleshooting AI-generated code

3. Python Environment and Reporting Setup

Installing and Configuring Python

1. Python installation
2. Package installation with pip
3. Managing Python environments
4. Verifying installations

Visual Studio Code Setup

1. Installing VS Code
2. Python extensions
3. Terminal usage
4. Running Python scripts

5. Productivity features

Jupyter Notebook Fundamentals

1. Creating notebooks
2. Running notebook cells
3. Markdown and documentation
4. Interactive reporting workflows

Installing Reporting Libraries

1. Installing Pandas
2. Installing Matplotlib
3. Installing Seaborn
4. Installing OpenPyXL
5. Verifying library compatibility

4. Working with Data Using Pandas

Introduction to Pandas

1. Understanding DataFrames
2. Understanding Series objects
3. Rows, columns, and indexes
4. DataFrame workflows

Importing Reporting Data

1. Reading CSV files
2. Reading Excel files
3. Working with worksheets
4. Handling delimiters and encodings
5. Import validation

Exploring and Understanding Data

1. Viewing data samples
2. Data inspection techniques

3. Data types and structures
4. Summary statistics
5. Understanding missing values

Data Cleaning and Preparation

1. Renaming columns
2. Removing duplicates
3. Handling missing values
4. Data formatting
5. String cleanup operations
6. Date formatting
7. Standardizing reporting data

Filtering and Sorting Data

1. Conditional filtering
2. Multiple conditions
3. Sorting records
4. Selecting subsets of data
5. Query-based filtering

Aggregation and Summarization

1. GroupBy operations
2. Summarizing metrics
3. Counting and averaging
4. Pivot table concepts
5. Reporting summaries

Combining and Transforming Data

1. Merging datasets
2. Joining DataFrames
3. Concatenating data
4. Calculated columns
5. Derived reporting metrics

Exporting Processed Data

1. Exporting CSV files
2. Exporting Excel files
3. Managing worksheets
4. Saving automated outputs

5. Data Visualization with Matplotlib and Seaborn

Understanding Reporting Visualizations

1. Importance of visual storytelling
2. Choosing the correct chart
3. Reporting presentation considerations
4. Avoiding misleading visualizations

Matplotlib Fundamentals

1. Creating basic charts
2. Line charts
3. Bar charts
4. Pie charts
5. Histograms
6. Scatter plots
7. Figure customization

Seaborn Fundamentals

1. Understanding Seaborn advantages
2. Built-in visualization themes
3. Statistical visualization concepts
4. Simplified chart creation

Building Business-Friendly Visuals

1. Styling charts professionally
2. Color palettes and readability

3. Labels and annotations
4. Multi-chart layouts
5. Dashboard-style reporting visuals

Visualizing Reporting Metrics

1. Trend analysis charts
2. Comparative reporting visuals
3. Distribution analysis
4. Correlation visualization
5. KPI-focused charts

Exporting Charts and Images

1. Saving charts as images
2. Embedding visuals into reports
3. High-resolution exports
4. Presentation-ready outputs

6. Excel Reporting Automation with OpenPyXL

Introduction to OpenPyXL

1. Understanding Excel automation
2. OpenPyXL architecture
3. Working with workbooks and worksheets
4. Reading and writing Excel files

Creating Automated Reports

1. Writing data into worksheets
2. Creating multiple sheets
3. Automating repetitive reporting tasks
4. Building reusable report templates

Formatting Excel Reports

1. Fonts and styles

2. Cell formatting
3. Borders and colors
4. Alignment and spacing
5. Conditional formatting

Working with Excel Features

1. Formulas and calculations
2. Merged cells
3. Filters and freeze panes
4. Tables and formatting structures
5. Data validation basics

Integrating Pandas with OpenPyXL

1. Exporting DataFrames into Excel
2. Post-processing exported reports
3. Combining charts and spreadsheets
4. End-to-end report generation workflows

Creating Presentation-Ready Outputs

1. Executive reporting layouts
2. Branded reporting concepts
3. Automated report packaging
4. Deliverable preparation

7. AI-Assisted Python Development for Reporting

Prompt Engineering for Reporting Tasks

1. Writing clear coding prompts
2. Explaining business requirements to AI
3. Requesting data transformation scripts
4. Requesting visualization scripts

Using ChatGPT to Generate Python Code

1. Generating Pandas workflows
2. Generating charting scripts
3. Generating Excel automation scripts
4. Iterative code refinement

Reviewing and Improving AI-Generated Code

1. Understanding generated logic
2. Validating outputs
3. Modifying generated scripts
4. Identifying incorrect assumptions

Practical AI-Assisted Reporting Workflows

1. Rapid report prototyping
2. AI-assisted debugging
3. AI-assisted documentation
4. Reusable prompt libraries

8. Building End-to-End Reporting Projects

End-to-End Reporting Workflow

1. Importing raw data
2. Cleaning and transforming datasets
3. Generating charts
4. Exporting polished Excel reports

Practical Reporting Scenarios

1. Sales reporting automation
2. Operational KPI reporting
3. HR reporting workflows
4. Financial summary reporting
5. Inventory and logistics reporting

Reporting Best Practices

1. File organization
2. Naming standards
3. Script readability
4. Reusability considerations
5. Maintaining reporting consistency

Productivity and Automation Strategies

1. Reducing repetitive work
2. Creating reusable templates
3. Combining Excel and Python workflows
4. Scaling reporting automation gradually

9. Operational Best Practices and Next Steps

Managing Reporting Projects

1. Organizing scripts and files
2. Versioning basics
3. Backup considerations
4. Workflow management

Troubleshooting and Maintenance

1. Common reporting automation issues
2. Handling broken datasets
3. Fixing formatting issues
4. Maintaining reusable scripts

Expanding Beyond Basic Reporting

1. Introduction to dashboards
2. Introduction to APIs
3. Introduction to scheduling automation
4. Future learning pathways

Continuing the Python Journey

1. Recommended practice approaches
2. Online learning resources
3. Building confidence with automation
4. Developing practical reporting portfolios

This training outline serves as a general framework intended to guide the delivery of the course content. The trainer reserves the right to modify, expand, reorganize, or adjust the topics, sequence, depth, practical exercises, tools, and delivery approach as deemed appropriate to accommodate participant backgrounds, operational requirements, technological updates, industry developments, and overall learning effectiveness without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur