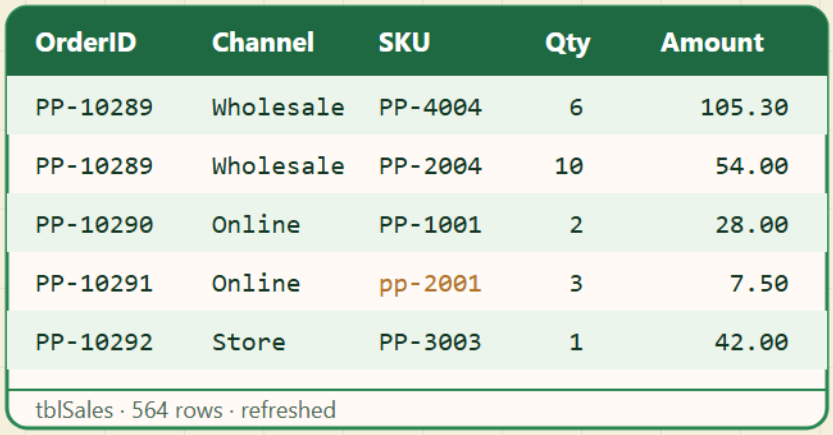
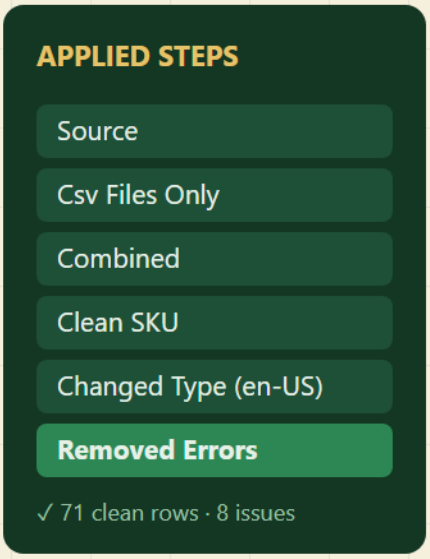
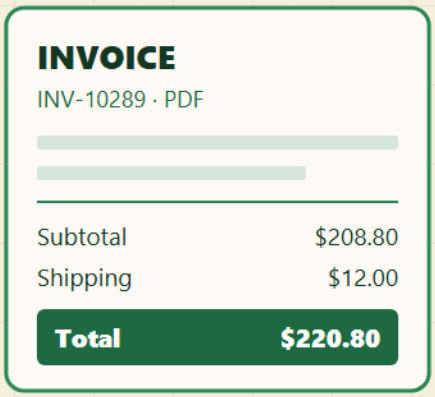


OrderID	Channel	SKU	Qty	Amount
PP-10289	Wholesale	PP-4004	6	105.30
PP-10289	Wholesale	PP-2004	10	54.00
PP-10290	Online	PP-1001	2	28.00
PP-10291	Online	pp-2001	3	7.50
PP-10292	Store	PP-3003	1	42.00

tblSales · 564 rows · refreshed




```
let
  Source = Folder.Files(pDataFolder),
  Typed = Table.TransformColumnTypes(...)
in Typed
```




```
Sub RunMonthEnd()
  RefreshAllData
  ExportInvoicePdf "PP-10289"
End Sub
```

A COMPLETE, PRACTICAL GUIDE

Excel Power Query and VBA Automation

Clean, transform, and automate your business data

Tables · XLOOKUP · Dynamic Arrays · LAMBDA · Power Query · M
PivotTables · Data Model · VBA · UserForms · PDF Invoices · Outlook

FORMULAS · POWER QUERY · REPORTING · VBA

THENEXBOOKS

Excel Power Query and VBA Automation

Clean, Transform, and Automate Your Business Data From messy CSV exports to a one-click month-end process — built step by step

First Edition

THENEXBOOKS

Excel Power Query and VBA Automation
Clean, Transform, and Automate Your Business Data

First Edition, 2026

Copyright © 2026 thenexbooks (thenexbooks.gumroad.com). All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means without the prior written permission of the publisher, except for brief quotations in reviews and certain other non-commercial uses permitted by copyright law. Formulas, Power Query M code, VBA code, and scripts in this book may be used freely in your own workbooks and projects.

The information in this book is provided "as is" without warranty of any kind. While every precaution has been taken in its preparation, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein. Microsoft Excel, Microsoft 365, and related products change over time; ribbon labels, dialogs, functions, and feature availability can differ between versions, update channels, and licences. Always consult Microsoft's documentation for the version you use, keep backups of your workbooks, and test automation on copies of your data before you rely on it.

Microsoft, Microsoft 365, Excel, Power Query, Power Pivot, Power BI, Power Automate, Outlook, OneDrive, SharePoint, Visual Basic, SQL Server, and Windows are trademarks of the Microsoft group of companies. Other product names are trademarks of their respective owners. Their use in this book is for identification only and does not imply endorsement. This book is independent and is not affiliated with, sponsored by, or endorsed by Microsoft Corporation.

Pebble & Pine Stationery, the shop used as the running example in this book, is fictional, as are its people, suppliers, wholesale customers, products, prices, and all sample data. Email addresses and web addresses use reserved example domains.

Preface

Every month, in thousands of offices and small businesses, someone opens the same set of files and does the same work again. A sales export arrives as a CSV file. Supplier price lists come in by email, each in a slightly different shape: one uses dollar signs, another writes dates the American way, a third adds a footer line that breaks every formula. Someone copies, pastes, trims, fixes, looks things up, builds a summary, prints invoices, and emails them — and next month, does it all again by hand, hoping nothing was missed.

Excel already contains everything needed to stop doing that work manually. **Tables and modern formulas** give your data structure and make calculations reliable. **Power Query** records every cleaning step once and replays it with a single click on next month's files. **VBA** automates the rest: generating documents, exporting PDF files, preparing emails, refreshing queries, and running a whole month-end process from one button. Used together, these tools turn a fragile afternoon of copy and paste into a dependable, repeatable process.

This book teaches all three, in the order you need them, using one realistic business from start to finish. We do not just tour the ribbon. Each feature appears at the moment it solves a real problem in the data, and we explain *why* it works, so you can apply the same ideas to your own files on Monday morning.

Who This Book Is For

- **Analysts and office professionals** who prepare the same reports every week or month and want them to refresh instead of being rebuilt.
- **Bookkeepers, operations staff, and small-business owners** who handle sales exports, price lists, inventory counts, and invoices in Excel.
- **Experienced Excel users** who know formulas well but have never opened the Power Query Editor or the VBA editor.
- **Anyone who inherits a macro-filled workbook** and needs to understand it, make it safer, or replace fragile parts with Power Query.

You should be comfortable with everyday Excel: entering data, writing a simple formula such as `=SUM(B2:B10)`, and creating a basic chart. No programming experience is required; Part V teaches VBA from the first line.

What You Will Learn

- How to structure business data with Excel tables and structured references, and how to clean it with text, number, and date formulas.

- How to look up and summarize with XLOOKUP, INDEX/MATCH, SUMIFS, and dynamic arrays such as FILTER, UNIQUE, and SORT, and how to make formulas readable with LET and reusable with LAMBDA.
- How to import CSV files, Excel workbooks, and whole folders with Power Query, and how to clean, reshape, merge, and append data with recorded, refreshable steps.
- How to read and write the M language, build parameters and custom functions, handle errors, check data quality, and keep refreshes fast and reliable.
- How to analyze results with PivotTables, slicers, timelines, the Data Model, simple DAX measures, charts, and dashboards.
- How to automate with VBA: the macro recorder, the language, the Excel object model, events, UserForms, error handling, arrays, and performance.
- How to work with files and folders, generate invoices and PDF files, prepare emails in Outlook, drive Power Query refreshes from code, secure macro-enabled workbooks, and understand where Office Scripts fit.

How This Book Is Organized

Part I — Foundations: Tables and Modern Formulas (Chapters 1–5) sets up the project and covers tables, data cleanup with formulas, lookups and conditional totals, dynamic arrays, LET, and LAMBDA. **Part II — Power Query Essentials** (Chapters 6–10) introduces the Power Query Editor and covers importing files and folders, cleaning and transforming, reshaping, and merging and appending queries. **Part III — Power Query in Depth** (Chapters 11–14) covers the M language, parameters and custom functions, error handling and data-quality checks, and refresh, load options, query folding, and performance. **Part IV — Analysis and Reporting** (Chapters 15–18) covers PivotTables, the Data Model and DAX measures, charts and dashboards, and a complete no-code reporting case study. **Part V — VBA Fundamentals** (Chapters 19–23) teaches VBA from the macro recorder to robust, fast, well-structured code. **Part VI — Automation in Practice** (Chapters 24–28) works with files and folders, invoices, PDF files and email, combines Power Query with VBA, covers security, sharing, and Office Scripts, and finishes with a capstone month-end automation. The appendices provide a cheat sheet, templates and reference material including the sample data script, a production checklist, and a glossary.

Each chapter includes hands-on steps, exercises under **Try It Yourself**, and a **Key Takeaways** box for quick review.

The Running Example: Pebble & Pine Stationery

Throughout the book we work with **Pebble & Pine Stationery**, a fictional small stationery business with a walk-in store, an online store, and a handful of wholesale customers such as a school, a library, and a coworking space. The shop sells 24 products — notebooks, pens, desk accessories, and art

supplies — from four suppliers. Every month it receives a sales export from its shop system and a price list from each supplier, and every month Sam, who runs operations and bookkeeping, has to turn those files into reports, reorder lists, and invoices.

The sample data covers January to March 2026: 564 sales lines, 12 supplier price-list files with realistic problems deliberately left in, a product master, inventory counts, and a budget. Appendix B contains a short PowerShell script that creates every sample file on your own computer in seconds, so your results match the book exactly. We start by organizing one month of sales in a table and finish with a month-end process that refreshes all data, checks its quality, creates invoices as PDF files, prepares emails, and exports a dashboard report — with one click.

Versions and Conventions

Examples target **Excel for Windows in Microsoft 365**. Most features also work in Excel 2021 and Excel 2024; where a function or feature needs a newer version, a Note says so and, where practical, we show an alternative that works in older versions. Power Query is built into Excel 2016 and later for Windows. VBA examples that use files, folders, and Outlook are written for Windows. Screens, ribbon labels, and dialogs can differ slightly between versions and update channels; if a command is not exactly where the book says, look nearby or use the **Search** box at the top of the Excel window.

The book assumes **US English regional settings**: dates appear as `3/18/2026` (or in ISO form, `2026-03-18`), currency as `$1,234.50`, and formulas use commas to separate arguments. If your computer uses different settings, Chapter 1 explains what changes and how to read the examples.

> **TIP**
> A recommended practice or a technique that makes your work easier.

> **NOTE**
> Background information, version differences, or a useful aside.

> **WARNING**
> A common mistake or something that can cause data loss, wrong results, or security problems.

Ribbon commands are written in bold with arrows, such as **Data > Get Data > From File > From Folder**. Formulas, Power Query M code, and VBA appear in `monospace`; long lines are wrapped to fit the page. Names follow a consistent convention — tables such as `tblSales`, queries such as `SupplierPrices`, parameters such as `pDataFolder`, functions such as `fnCleanSKU`, and VBA modules such as `modInvoices` — summarized in Appendix B.

Contents

Preface

PART I — FOUNDATIONS: TABLES AND MODERN FORMULAS

1 Excel for Business Data: Versions, Workbooks, and the Pebble & Pine Project

2 Excel Tables and Structured References

3 Clean Data with Formulas: Data Types, Validation, Text, and Dates

4 Lookups and Conditional Totals: XLOOKUP, INDEX/MATCH, and SUMIFS

5 Dynamic Arrays, LET, and LAMBDA: FILTER, UNIQUE, SORT, and Reusable Formulas

PART II — POWER QUERY ESSENTIALS

6 Meet Power Query: Get Data, the Editor, and Applied Steps

7 Importing Data: CSV Files, Excel Workbooks, and Whole Folders

8 Cleaning and Transforming: Data Types, Text, Dates, and Replacements

9 Reshaping Data: Unpivot, Pivot, Group By, and Custom Columns

10 Combining Data: Merge Queries and Append Queries

PART III — POWER QUERY IN DEPTH

11 The M Language: let Expressions, Values, Lists, Records, and Tables

12 Parameters, Custom Functions, and Reusable Queries

13 Error Handling and Data Quality Checks

14 Refresh, Load Options, Query Folding, and Performance

PART IV — ANALYSIS AND REPORTING

15 PivotTables: Summaries, Grouping, Slicers, and Timelines

16 The Data Model: Relationships, Measures, and a First Look at DAX

17 Charts and Dashboards for Business Data

18 Case Study: A No-Code Monthly Sales and Margin Report

PART V — VBA FUNDAMENTALS

19 Getting Started with VBA: The Macro Recorder, the Editor, and Macro-Enabled Workbooks

20 VBA Language Essentials: Variables, Procedures, Decisions, Loops, and Debugging

21 The Excel Object Model: Workbooks, Worksheets, Ranges, and Tables

22 Events and UserForms: Responsive Workbooks and Simple Data-Entry Screens

23 Robust VBA: Error Handling, Arrays, and Performance

PART VI — AUTOMATION IN PRACTICE

24 Working with Files and Folders: Dir, FileSystemObject, and Batch Processing

25 Invoices, PDFs, and Email: Generating and Sending Business Documents

26 Power Query Meets VBA: Refresh Automation, Parameters, and Logging

27 Security, Sharing, and Office Scripts

28 Capstone: The Pebble & Pine Month-End Automation

APPENDICES

A Excel Power Query and VBA Cheat Sheet

B Templates, Reference, and the Sample Data Script

C Production Checklist

D Glossary

PART I

Foundations: Tables and Modern Formulas

Setting up the Pebble & Pine project, giving data structure with Excel tables and structured references, cleaning text, numbers, and dates with formulas, looking up and summarizing with XLOOKUP and SUMIFS, and building spill-based reports with dynamic arrays, LET, and LAMBDA.

1 · Excel for Business Data

2 · Excel Tables and Structured References

3 · Clean Data with Formulas

4 · Lookups and Conditional Totals

5 · Dynamic Arrays, LET, and LAMBDA

CHAPTER 1 · PART I — FOUNDATIONS: TABLES AND MODERN FORMULAS

Excel for Business Data: Versions, Workbooks, and the Pebble & Pine Project

Before we clean, transform, or automate anything, we need a clear picture of the job, the tools, and the data. In this chapter you meet the small business that runs through the whole book, set up its folders on your PC, generate the sample files, and build the first version of the workbook we will grow chapter by chapter.

What "Business Data Work" in Excel Really Looks Like

Most people who use Excel at work are not building financial models from scratch. They are moving data from one place to another and making it fit. A sales system exports a CSV file. A supplier emails a price list. Someone keeps a product list in a workbook that only they understand. At the end of every month, a person sits down, opens all of those files, copies the useful parts into a report, fixes whatever looks wrong, and sends the result to the boss. Next month they do it all again.

This copy, paste, and fix cycle is so common that it rarely gets a name, but it has predictable problems:

- **It is slow.** Every step is manual, so the work grows with the number of files and rows. A task that takes an hour with three files takes a day with thirty.
- **It hides errors.** A product code with an extra space, a price typed as text, or a date read the wrong way round does not shout at you. It silently drops out of a total, and the report looks fine.
- **It leaves no trail.** When the total is wrong, nobody can say which of last month's forty manual edits caused it, because the edits were never written down.
- **It depends on one person.** The process lives in someone's head. When that person is on vacation, the report is late or wrong.

The goal of this book is to replace that cycle with a process that is **repeatable** (the same steps run the same way every month), **visible** (you can see and check each step), and **mostly automatic** (you drop new files into a folder and press a button). We get there in stages, and every stage is useful on its own. Even if you stop after Part I, your workbooks will be cleaner and your totals more trustworthy.

The Three Tools of This Book

Excel gives you three very different ways to work with business data. Knowing which one to reach for is half the skill, so we introduce all three now and use them in order through the book.

1. Tables and formulas (Part I)

An Excel **table** is a range of data that Excel knows about: it has a name, a header row, and columns that grow automatically as you add rows. Formulas that refer to tables read like sentences, for example `=SUM(tblSales[Amount])`. Modern lookup and summary functions such as XLOOKUP and SUMIFS, and dynamic array functions such as FILTER and UNIQUE, let you build live reports that update the moment the data changes. Tables and formulas are the foundation of everything else: Power Query loads into tables, and VBA works best with tables.

2. Power Query (Parts II–IV)

Power Query (labeled **Get & Transform Data** on the Data tab) is a tool for importing and reshaping data. You connect to a file, a folder, or a database, then apply cleaning steps: remove blank rows, fix data types, trim spaces, merge with a lookup list, and so on. Power Query records every step. Next month you do not repeat the work; you click **Refresh** and the same steps run on the new data. Power Query never changes your source files, which makes it safe to experiment with.

3. VBA (Parts V–VI)

VBA (Visual Basic for Applications) is the programming language built into desktop Excel. It can do things that formulas and Power Query cannot: create files, save PDFs, move files between folders, react when someone edits a cell, show a data-entry form, and draft emails in Outlook. We use VBA for the "last mile" of automation, the actions around the data rather than the data cleaning itself.

Which tool for which job?

The table below is a rule of thumb we follow throughout the book. When two tools could do a job, we prefer the one higher on the list, because it is easier for the next person to understand and maintain.

Task	First choice	Why
Calculate a value per row (Amount = Qty × price)	Formula in a table	Instant, visible, recalculates by itself
Look up a product name or a price	Formula (XLOOKUP, INDEX/MATCH)	Live link between two tables
Summarize by channel, category, or month	SUMIFS or a PivotTable	No code; easy to audit
Import CSV files and fix their problems	Power Query	Steps are recorded and repeat on refresh
Combine every file in a folder	Power Query	New files are picked up automatically
Join sales with products or customers	Power Query (Merge)	Handles thousands of rows cleanly

Create PDFs, move files, draft emails	VBA	Formulas and queries cannot act on files
Run a whole month-end with one click	VBA calling Power Query	Orchestrates the other tools

Notice what is missing: we never use VBA to clean data that Power Query could clean, and we never write a formula that copies data from one place to another by hand. Each tool does the job it is best at.

Excel Versions and What They Include

"Excel" is not one product. Several versions are in use at the same time, and they differ in which functions they offer. This book targets **Excel for Windows in Microsoft 365** (the subscription version, which receives new features regularly). Most of the book also works in the perpetual, buy-once versions **Excel 2021** and **Excel 2024**, and we point out the exceptions as we go.

To see which version you have, choose **File > Account** and look under **Product Information**; the **About Excel** button shows the exact version and build. The table summarizes the features this book relies on.

Feature	Microsoft 365	Excel 2024	Excel 2021	Excel 2016/2019
Tables, SUMIFS, INDEX/MATCH	Yes	Yes	Yes	Yes
Power Query (Get & Transform)	Yes	Yes	Yes	Yes
XLOOKUP, FILTER, UNIQUE, SORT, LET	Yes	Yes	Yes	No
LAMBDA, MAP, BYROW, VSTACK, TEXTSPLIT	Yes	Yes	No	No
GROUPBY, PIVOTBY, REGEX functions	Recent builds	No	No	No
VBA macros	Yes	Yes	Yes	Yes

Whenever a chapter uses a function that older versions lack, we also give a **fallback formula** built from classic functions such as INDEX, MATCH, and SUMIFS. If you share workbooks with colleagues on older versions, use the fallback; a formula that shows `#NAME?` on someone else's PC is worse than no formula at all.

Windows first; a word about Mac and the web

The book is written for Excel on Windows. **Excel for Mac** includes Power Query, but with fewer data connectors, and some VBA features that depend on Windows (Outlook automation, the FileSystemObject) do not work there. **Excel for the web** can open the workbooks we build and can refresh some queries, but it cannot author most Power Query steps and it does not run VBA at all. You can view results on a Mac or in a browser; build them on Windows.

> **NOTE**
>
> Ribbon labels and menu names may differ slightly between versions and update channels. If a command is not exactly where we describe it, look for the closest label on the same tab, or type the command name into the **Search** box at the top of the Excel window.

Workbook File Types

The file type you choose when saving decides what the workbook can contain. Four types matter in this book.

Extension	What it is	Use it for
`.xlsx`	Standard workbook; cannot store macros	Data, formulas, Power Query, PivotTables
`.xlsm`	Macro-enabled workbook	Any workbook that contains VBA code
`.xlsb`	Binary workbook; can store macros	Very large workbooks, the Personal Macro Workbook
`.csv`	Plain text, one sheet, values only	Exchanging data between systems

A **CSV file** ("comma-separated values") is just text: one line per row, with commas between the fields. It has no formatting, no formulas, no tables, and only one sheet. Business systems love CSV because every program can read it. We will treat CSV files as **inputs**, never as places to keep our work.

> **WARNING**
>
> Never press Ctrl+S on a CSV export you have opened in Excel and edited. Excel rewrites the file in its own way: formulas become values, formatting is lost, and dates and numbers may be written differently from the original export. Keep source files exactly as you received them and save your work in a real workbook (`.xlsx` or `.xlsm`) instead.

Meet Pebble & Pine Stationery

Our running example is **Pebble & Pine Stationery**, a small, fictional shop at 214 Willow Street, Portland, OR 97205. It sells notebooks, pens, paper, desk accessories, and art supplies through three channels:

- **Store**: the walk-in shop, open Monday to Saturday and closed on Sundays.
- **Online**: the web shop at pebbleandpine.example.com, which takes orders every day.
- **Wholesale**: a handful of accounts (a school, a library, an architecture office, a coworking space, and a day care) that buy in bulk at 10% off retail and receive invoices.

The owner, **Nora Lindqvist**, wants a clear monthly picture: how much did we sell, through which channel, in which category, at what margin, and what do we need to reorder? The person who produces that picture is **Sam Ortiz**, who handles operations and bookkeeping. In many scenarios in this book, you are Sam.

The data Sam works with

Every month Sam deals with the same kinds of files. Here is the full set we use in the book, all for the first quarter of 2026:

File	Contents	Rows
`Sales\Sales_2026-01.csv` (and -02, -03)	One line per product sold, exported by the shop system	185 / 183 / 196
`PriceLists\SUP01_2026-01.csv` … (12 files)	Monthly supplier price lists, four suppliers	79 data rows in total
`Products.csv`	The product master: SKU, name, category, supplier, retail price, reorder level	24
`Suppliers.csv`	Supplier names, emails, payment terms	4
`Customers.csv`	Wholesale customers and their addresses	5
`Inventory.csv`	Stock count on March 31, 2026	24
`Budget_2026.csv`	Monthly sales budget by category	5

Across the three months there are **564 sales lines**. That is small enough to inspect by eye when we need to and large enough that manual fixing is painful, which makes it a realistic training set.

Sam's monthly pain points

Before this book, month-end at Pebble & Pine looks like this:

1. Open the month's sales export, add an Amount column, and fix any product codes that do not match the product list.
2. Copy the result into last month's report workbook and hope the formulas still point at the right rows.
3. Open four supplier price lists, each in a slightly different format: one has blank rows, one writes prices with a dollar sign, one uses lowercase codes, and one writes dates the American way and adds a footer line.
4. Work out which products have fallen below the shop's target margin of 55% and which need reordering.
5. Type each wholesale order into an invoice template, save it as a PDF, and attach it to an email.

By the end of the book, every one of these steps is either a refreshable query or a button. Chapter 28 runs the entire March month-end with one macro. We start much more modestly: by getting the files onto your PC.

Set Up the Project Folders

All the examples in the book use the same folder layout on drive C. Using exactly these paths will let you copy formulas, queries, and code from the book without editing them. (Chapter 12 shows how to make the paths configurable, so you are not locked into drive C forever.)

Folder layout

```
C:\PebblePine\                  workbooks live here
    PebblePine_Master.xlsx      (built in Chapters 1-2)
    Data\                       inputs: never edit these by hand
        Sales\                  Sales_2026-01.csv ... Sales_2026-03.csv
        PriceLists\             SUP01_2026-01.csv ... SUP04_2026-03.csv
        Products.csv, Suppliers.csv, Customers.csv,
        Inventory.csv, Budget_2026.csv
    Output\
        Invoices\               invoice PDFs (Part VI)
        Reports\                report PDFs (Part VI)
    Archive\                    processed files are moved here (Part VI)
```

You can create the folders in File Explorer (**Home > New folder**, or right-click > **New > Folder**), or let Windows PowerShell create them in one go:

1. Click **Start**, type `PowerShell`, and open **Windows PowerShell**. You do not need to run it as administrator.
2. Type the following command on one line (it is shown on two lines here only to fit the page) and press Enter:

Windows PowerShell

```
New-Item -ItemType Directory -Force -Path C:\PebblePine\Data, C:\PebblePine\Archive,
    C:\PebblePine\Output\Invoices, C:\PebblePine\Output\Reports
```

PowerShell lists the folders it created. Running the command a second time does no harm; `-Force` simply accepts folders that already exist. The script in the next section creates `Data\Sales` and `Data\PriceLists` for you.

Generate the Sample Files

The sample data is produced by a short PowerShell script, `make_data.ps1`, printed in full in **Appendix B**. It needs nothing beyond the PowerShell that comes with Windows, it downloads nothing, and it always produces exactly the same files, so your numbers will match the book to the cent.

1. Open **Notepad** and type (or paste) the script from Appendix B. It is about 200 lines. Take care with commas, quotes, and brackets.
2. Choose **File > Save As**. Browse to `C:\PebblePine\`, set **Save as type** to **All files (*.*)**, and enter the file name `make_data.ps1`. Keep the encoding as UTF-8 and click **Save**.
3. In the PowerShell window, run:

Windows PowerShell

```
powershell -ExecutionPolicy Bypass -File C:\PebblePine\make_data.ps1
```

After a second or two you should see one line of output:

Result

```
Sample files written to C:\PebblePine\Data (564 sales lines)
```

Open `C:\PebblePine\Data` in File Explorer and check that you have 21 files: five CSV files directly in `Data`, three in `Sales`, and twelve CSV files plus `_notes.txt` in `PriceLists`. That small text file is there on purpose; it will teach us something about folder imports in Chapter 7.

> **TIP**
>
> If Notepad saves the script as `make_data.ps1.txt`, Windows is hiding file extensions. In File Explorer choose **View > Show > File name extensions** (on Windows 10: **View > File name extensions**), rename the file to remove `.txt`, and run the command again. If PowerShell reports an error with a line number, compare that line with Appendix B character by character; a missing quote or bracket is the usual cause.

> **NOTE**
>
> The `-ExecutionPolicy Bypass` switch applies only to this one run of PowerShell. It does not change any setting on your PC. If your organization blocks scripts entirely, ask your IT team to run the script for you, or to copy the 21 generated files to a folder you can reach.

Build the Master Workbook

Now we create the first workbook of the project, `PebblePine_Master.xlsx`. Throughout Part I it holds the January sales and the reference lists as Excel tables. Later, Power Query reads from it. We start with the January sales export.

1. In Excel choose **File > Open > Browse**, go to `C:\PebblePine\Data\Sales`, change the file filter next to the file name box to **All Files (*.*)** or **Text Files**, and open `Sales_2026-01.csv`. (Double-clicking the file in File Explorer also works if CSV files open in Excel on your PC.)

2. Excel shows the data on a sheet named `Sales_2026-01`, after the file. Double-click the sheet tab, type `Sales`, and press Enter.
3. Choose **File > Save As > Browse**. Go to `C:\PebblePine\`, type the file name `PebblePine_Master`, and in **Save as type** pick **Excel Workbook (*.xlsx)**. Click **Save**.

The title bar now shows `PebblePine_Master.xlsx`. From this point on, the CSV file and the workbook are separate: the CSV stays untouched in `Data\Sales`, and our work happens in the workbook.

> **WARNING**
>
> Step 3 matters. If you skip the **Save as type** choice, Excel may keep the CSV format, and everything you build in the next chapters (tables, formulas, formatting) will be thrown away when you close the file. Always confirm that the title bar ends in `.xlsx`.

A First Look at the Data

Before changing anything, take five minutes to understand what you have. Good data work starts with looking. The sheet has seven columns and 186 rows: one header row and 185 data rows. Each data row is one **line**, meaning one product on one order.

Column	Example (row 2)	Meaning
A OrderID	PP-10001	Order number; one order can have several lines
B OrderDate	1/1/2026	Date of the order (ISO `2026-01-01` in the file)
C Channel	Store	Store, Online, or Wholesale
D CustomerID	(blank)	Filled only for wholesale lines, for example C001
E SKU	PP-4006	Product code (stock keeping unit)
F Qty	2	Units sold; negative for a return
G UnitPrice	22	Price per unit in dollars (22.00 in the file)

The first line reads: order PP-10001 on January 1, 2026, in the store, two units of PP-4006 (the Mixed Media Pad A3) at $22.00 each. Excel displays `22` rather than `22.00` because a CSV carries no number formats; the value is the same, and we will apply a currency format in Chapter 2.

Quick checks with the status bar

The status bar at the bottom of the Excel window is the fastest audit tool you have. Select a range and it shows the Average, Count, and Sum of the selection. Right-click the status bar to switch on more statistics, such as Numerical Count, Minimum, and Maximum.

- Click the column letter **F** to select the Qty column. The status bar shows **Sum: 443**, the total units sold in January, and a Count of 186 (Count includes the header text).

- Select **A2:A186**. Count shows **185**, the number of lines.
- Select column **D**. Count shows 7: the header plus six filled cells, the six wholesale lines.

Lines are not orders

Look at rows 3 and 4: both belong to order PP-10002, an online order for two different products. A report that says "185 sales" would be wrong; January has 185 **lines** but only **122 orders**. Keeping the two ideas separate matters for averages: the average order value (sales divided by orders) is very different from the average line value. We will count distinct orders properly in Chapter 4.

Problems hiding in plain sight

This export is typical: mostly clean, with a few traps that only show up when totals do not match. Scroll through and you can find them:

- **A bad product code.** Cell E65 (order PP-10038) contains `pp-2001`, with a leading space and lowercase letters, instead of `PP-2001`. To a person it is obviously the same product; to a lookup formula it is a different string. It is easy to miss because the space is invisible. Click E65 and look at the formula bar, or type `=LEN(E65)` in an empty cell: the result is 8 instead of 7.
- **Returns.** Rows 101 and 173 (orders PP-10337 and PP-10338) have a quantity of `-1`. They are refunds in the store. They are correct data, not errors, but any report must decide how to treat them. In this book, reports show **net sales**, so returns are included as negative amounts.
- **Order numbers out of sequence.** The two returns have order numbers higher than every January sale, because the shop system numbers refunds separately. Never assume that order numbers tell you the date.
- **Blank customer IDs.** 179 of 185 lines have no CustomerID. That is expected (store and web customers are anonymous), but it means CustomerID cannot be used to count customers.
- **No Amount column.** The export gives quantity and unit price but not the line amount. We add it in Chapter 2.

None of these problems is dramatic, and that is exactly why they are dangerous. The bad SKU is one line out of 185, worth $12.50. If you look up product names with a formula, that line shows an error or a blank, and the category totals come out $12.50 short. Chapter 3 fixes it with a formula, and Part II fixes it permanently with Power Query (Chapters 6 and 7).

Regional Settings: Separators and Dates

Excel takes several behaviors from the Windows regional settings of the PC it runs on, and this affects how formulas look and how CSV files open.

- **Argument separators.** This book writes formulas the way they appear on a PC with US settings: commas between arguments and a period as the decimal point, for example `=ROUND(A2*0.9, 2)`. In many countries Excel uses semicolons instead, so you would type `=ROUND(A2*0,9; 2)`. If Excel

rejects a formula from the book, replace the commas between arguments with semicolons. Function names stay in English only if your Excel is in English; localized versions translate them.

- **Dates in CSV files.** When Excel opens a CSV file directly, it interprets dates using your regional settings. The Pebble & Pine sales files use the ISO format `2026-01-08`, which Excel usually recognizes everywhere. Other files are less friendly: one supplier writes `3/1/2026`, meaning March 1. On a PC set to day-month order, Excel reads that as January 3 or leaves it as text. Power Query lets you state the file's format explicitly, which is one of the main reasons we use it (Chapter 8).
- **Display formats.** The book shows dates as `3/18/2026` or `2026-03-18` and money as `$1,234.50`. Your screen may show the same values in your local style. The underlying numbers are identical.

A quick test tells you whether dates arrived as real dates: select **B2:B186** and look at the status bar. If it shows a Sum, Excel sees numbers, because dates are stored as numbers (Chapter 3 explains this). If it shows only a Count, the dates are text, and you should re-open the file with Power Query once you reach Chapter 6.

Conventions Used in This Book

A few conventions make the instructions compact and unambiguous:

- **Ribbon paths** are written in bold with arrows, for example **Data > Get Data > From File > From Text/CSV**: tab, then group or button, then menu item.
- **Things you type**, file names, formulas, and code appear in `monospace`. Longer formulas, queries, and VBA code appear in boxes with a caption that says where they go, such as *Formula (tblSales, column Amount)*.
- **Names** follow fixed patterns. Worksheet tables start with `tbl` (`tblSales`), Power Query parameters with `p` (`pDataFolder`), custom query functions with `fn` (`fnCleanSKU`), and VBA modules with `mod` (`modInvoices`). Use the same names and every later chapter will fit your workbook.
- **Results** are shown as plain text tables in US formats, so you can compare them with your screen.
- **Callouts**: a *Tip* is a shortcut or better way, a *Note* adds background, and a *Warning* flags something that can lose data or produce wrong numbers.
- Every chapter ends with **Try It Yourself** exercises and **Key Takeaways**.

Each part of the book builds on the files from the previous part. Part I works in `PebblePine_Master.xlsx`. Parts II–IV build a separate Power Query workbook, `PebblePine_Reports.xlsx`, which reads from the master. Parts V and VI turn that report workbook into a macro-enabled automation workbook. Keep your files; you will need them.

When Data Outgrows Excel

Excel is the right home for Pebble & Pine's data: a few hundred lines a month, a handful of reference lists, and one person maintaining them. A worksheet holds up to 1,048,576 rows, and the Data Model in Chapter 16 can hold more, but size is not the only limit. If many people need to enter data at the same time, or you need strict rules about who can change which records, a database is a better fit; our companion book *Microsoft Access from Zero to Production* covers that step. The skills from this book carry over, because Power Query connects to databases just as easily as to CSV files.

Try It Yourself

1. Open `Sales_2026-02.csv` in Excel (do not save it). Use the status bar to find the number of lines and the total units for February. You should see 183 lines and 414 units. Close the file without saving.
2. In `PebblePine_Master.xlsx`, filter or scroll the Channel column and find the two wholesale orders in January. Which customer IDs do they belong to, and on which dates were they placed? (Answer: C001 on January 8 and C002 on January 22.)
3. Find the largest single line in January by sorting a *copy* of the sheet by Qty, largest first, and note the order number. Then close the copy without saving. (Hint: it is a wholesale line of 40 units.)
4. Open `PriceLists\SUP02_2026-02.csv` in Notepad, not Excel. Find the product whose cost is written as `N/A` and note how the prices are written. Close Notepad without saving.
5. Check your regional settings: in an empty cell of the master workbook type `=DATE(2026,3,18)` and press Enter. Note how your PC displays it. Then delete the cell.