Drive Folder Report — Explanation

Detailed breakdown of what each function does and how the treeview visualization and statistics collection work together to produce a complete folder status report.

The Problem

Google Drive is used heavily as backup. Even with careful organization, managing large quantities of folders and files becomes complex. Manually clicking through every directory to check status, count files, identify large files, or see what was recently modified takes hours.

The script solves this by scanning everything automatically and producing a single report that shows the complete picture in seconds.

What Each Function Does

format_size()

Converts raw byte counts into human-readable format. Instead of seeing "1073741824 bytes", you see "1.00 GB". Makes the report instantly understandable without mental math.

generate_treeview()

Creates a complete visual representation of the entire folder hierarchy — every folder, every subfolder, every file. Uses ASCII characters (├── └── │) to draw the tree structure. Shows folders with 📁 icons and files with 📄 icons. Fully recursive with no limits — scans everything from root to the deepest nested folder. You get the complete picture of the entire structure in one view.

scan_folder()

Walks through every folder and file recursively. For each file, it collects: name, full path, size in bytes, last modified timestamp, and file extension. Aggregates everything into statistics: total counts, size totals by extension, and identifies the 10 largest and 10 most recently modified files.

generate_report()

Assembles all collected data into a structured TXT report. Organizes information into clear sections: Summary, Folder Structure (treeview), Files by Type, Largest Files, and Recently Modified. Easy to scan, easy to archive.

create_report()

Main entry point. Takes a folder path, runs all the analysis, generates the report, and saves it to a file. One function call produces the complete report.

Treeview Visualization

The treeview gives you immediate understanding of folder organization without opening a file explorer. Example output:

```
📁 Trading/
├── 📁 Backtest/
|   ├── 📁 Results/
|   |   ├── 📄 EURUSD_2024_Q1.csv
|   |   ├── 📄 EURUSD_2024_Q2.csv
|   |   ├── 📄 EURUSD_2024_Q3.csv
|   |   ├── 📄 EURUSD_2024_Q4.csv
|   |   ├── 📄 GBPUSD_2024_Q1.csv
|   |   ├── 📄 GBPUSD_2024_Q2.csv
|   |   └── 📄 GBPUSD_2024_Q3.csv
|   └── 📁 Scripts/
|       ├── 📄 backtest_runner.py
|       ├── 📄 analysis.py
|       └── 📄 report_generator.py
├── 📁 Data/
|   ├── 📄 EURUSD_1H.csv
|   ├── 📄 GBPUSD_1H.csv
|   ├── 📄 USDJPY_1H.csv
|   ├── 📄 AUDUSD_1H.csv
|   └── 📄 NZDUSD_1H.csv
└── 📁 Reports/
    ├── 📄 weekly_summary.pdf
    ├── 📄 monthly_report.pdf
    └── 📄 yearly_overview.pdf
```

Every single folder, every single file — nothing hidden, nothing truncated. At a glance you see the complete picture. What would take 10 minutes of clicking takes 2 seconds to read.

Report Sections

Summary

Quick numbers: total folders, total files, total size. Instant overview of the scope.

Folder Structure

The treeview visualization. See the complete hierarchy at once.

Files by Type

Breakdown by file extension. See how many CSV files exist, how much space PDFs take, which file types dominate. Useful for understanding what the folder actually contains.

Largest Files

Top 10 biggest files. Immediately identify what's consuming the most space. Essential for cleanup decisions.

Recently Modified

Top 10 most recently changed files. Shows what's been actively worked on. Useful for tracking activity and finding recent work.
