

# EAG Brief 9 — Pakistan Sectoral Linkages

## Replication package

---

This folder reproduces, from raw data, the two figures in **EAG Brief 9**:

- `pakistan_decomposition.png` — where each Pakistani sector's gross output value originates (domestic value added by source sector, imported inputs, and net taxes).
- `pakistan_minus_developingasia.png` — the same, expressed as Pakistan **minus** the developing-Asia benchmark.

Everything needed to regenerate the figures is here; nothing outside this folder is required.

## Please cite

If you use these figures, the underlying decomposition, or this code in your own work, please cite **EAG Brief 9**:

Pirzada, A.J. (2026) *Pakistan's Economy: Sectoral Linkages and Comparison with Developing Asia*, EAG Brief No. 9. Economic Advisory Group. Available at: [https://eag.org.pk/eag-briefs/#flipbook-df\\_1321/1/](https://eag.org.pk/eag-briefs/#flipbook-df_1321/1/) (Accessed: 4 July 2026).

## Folder structure

```
EAG Brief 9 Replication/
|-- va_decomposition_workflow.py    The single script that does everything.
|-- README.pdf                     This file.
|-- Country Tables/                 Input data: 25 ADB national IO tables (.xlsx).
|-- Theory/
|   |-- decomposition_algebra.tex   Matrix algebra behind the method (LaTeX source).
|   '-- decomposition_algebra.pdf   ...compiled. Appendix A: exact 35->15 mapping.
|-- EAG Brief/
|   |-- EAG Brief 9 - Pakistan Sectoral Linkages.docx
|   '-- EAG Brief 9 - Pakistan Sectoral Linkages.pdf
|
|   -- generated by the script (safe to delete; re-created each run) --
|-- country_decompositions.xlsx     One tab per country: its decomposition matrix.
|-- pakistan_minus_developing_asia.xlsx  Pakistan, benchmark median, difference.
|-- pakistan_decomposition.png        Figure 1.
'-- pakistan_minus_developingasia.png   Figure 2.
```

## How to replicate

**Requirements:** Python 3 with numpy, pandas, matplotlib, openpyxl. Install once if needed:

```
pip install numpy pandas matplotlib openpyxl
```

**Run** from inside this folder:

```
python va_decomposition_workflow.py
```

On Windows, if `python` opens the Microsoft Store, use the launcher instead:

```
py -3 va_decomposition_workflow.py
```

That single command runs three sections in order and rewrites all four generated files above:

1. **Decompose every country.** Loops over `Country Tables/`, applies the standard Leontief-inverse decomposition to each, and writes one tab per country to `country_decompositions.xlsx`.
2. **Build the comparison.** Reads that workbook back, takes the cell-wise median across the 18 developing-Asia economies, subtracts it from Pakistan, and writes `pakistan_minus_developing_asia`.
3. **Draw the figures.** Reads the spreadsheets and writes the two `.png` files.

The run prints the benchmark economies used and the files written; it takes under a minute.

## Where things are defined

- **The method** (production function, accounting identity, the algebra, and the 35→15 sector mapping) is documented in `Theory/decomposition_algebra.pdf`.
- **Which countries form the “developing Asia” benchmark** is the `COUNTRIES` table near the top of `va_decomposition_workflow.py`, with a one-line reason next to each economy that is excluded (developed, outlier, or Pakistan itself).
- **The 35→15 sector mapping** the code uses is the `MAP` dictionary and `PAPER` list in the script, tabulated in Appendix A of the Theory PDF.

## Notes

- The figures the brief shows use the **15-sector** mapping (which splits manufacturing into Textiles; Petrol. & Chem.; Machinery & Elec.; Transport eq.; Other mfg). This is a refinement of the 11-sector aggregation used elsewhere in the paper.
- To regenerate the Theory PDF after editing its source, run `pdflatex decomposition_algebra.tex` (twice) from the `Theory/` folder.
- Source of the input tables: ADB Input–Output Tables (2022 release), sheet **Table 1.17**.