Replication Guide for "Effects of Subsidies on Welfare and Market Structure in the U.S. Broadband Industry"

Sidi Mohamed Sawadogo

June 2026

1 Purpose

This guide gives the run order for reproducing the 2020 structural demand, fixed-cost, and counterfactual results used in the paper. It records the main inputs, scripts, outputs, and verification checks. The methodological details are in the paper and appendix.

The pipeline first constructs the structural dataset and CSS broadband subscription shares. It then estimates nested-logit demand and marginal costs, constructs Fan–Yang profit bounds, estimates a seven-parameter fixed-cost model, builds the tract-level FPL eligibility file, draws 50 fixed-cost vectors from the confidence region, and simulates baseline, ACP, and BEAD portfolio best-response outcomes.

The package produces draw-level files and trimmed interval tables for welfare, entry, prices, quantities, and concentration. The baseline, ACP, and BEAD simulations use the same fixed-cost draw index. Draw-level comparisons should therefore match on `draw`.

2 Folder Structure

The author-side working folder is:

```
/Users/sidimohamedsawadogo/Package thesis/JMP/Code
```

Any local path is fine, but scripts should be run from the local equivalent of the `Code` folder. The intended package structure is:

Folder	Contents
`data/`	Input data, intermediate CSV files, final CSV tables, and run logs.
`graphic/`	PDF and PNG figures.
`documentation/`	TeX and PDF documentation, including this guide.
`Archives/`	Archived or alternative scripts not needed for the main run.

Some older scripts still write selected intermediate files to the root of `Code`. In the archived package, CSV outputs belong in `data/`, figures in `graphic/`, and TeX/PDF documentation in `documentation/`.

3 Before Running the Pipeline

Before running the full sequence, check:

- Run every script from the `Code` folder, not from `data/`, `graphic/`, or `documentation/`.

- Run script 00 before script 01 or script 02 if the structural input files need to be rebuilt.
- Run script 01 after script 00 if the CSS subscription-share inputs need to be rebuilt.
- Run script 05 before script 07 if the ACS FPL eligibility file needs to be rebuilt.
- `data/finaldataset.csv` is present. This is the 2020 structural dataset and the upstream input for the demand stage.
- `data/acs_tract_2020_fpl.csv` is present before running the ACP counterfactual. Script 07 uses it to construct eligible population shares.
- Use Python 3.9 or later. The package was tested on Python 3.14.
- Do not mix outputs from different runs of the fixed-cost stage. Scripts 06–09 must use the same `fixed_cost_simulation_draws_2020.csv`.

4 Execution Order

Run the scripts in the order below. Script 00 constructs the upstream structural files. Script 01 constructs CSS subscription shares. The current package ships `data/finaldataset.csv`; if it is regenerated, rebuild it after scripts 00–01 and before script 02. Script 02 creates the demand and marginal-cost primitives used later. Scripts 03–04 construct the Fan–Yang bounds and fixed-cost draws. Script 05 builds the ACS FPL eligibility input used by the ACP counterfactual. Scripts 06–08 solve the baseline and policy counterfactuals. Script 09 combines the draw-level outputs into the cost-benefit analysis.

1. `00. build_structural_dataset_2020.py`
2. `01. css_simple_logit_summary_2020.py`
3. `02. nested_logit_demand_supply_2020.py`
4. `03. estimate_profit_bounds_fan_yang_2020.py`
5. `04. compute_fixed_cost_fan_yang_2020.py`
6. `05. build_acs_2020_fpl.py`
7. `06. simulate_baseline_portfolio_br_outcomes_2020.py`
8. `07. simulate_acp_portfolio_br_nash_counterfactual_2020.py`
9. `08. simulate_bead_portfolio_br_nash_counterfactual_2020.py`
10. `09. cost_benefit_analysis_2020.py`

Run the full sequence only after the required inputs are in place. If all upstream inputs already exist and the goal is only to regenerate counterfactuals, start from script 06. If the fixed-cost model or draw selection changes, rerun scripts 06–09 in order.

Smoke Tests

Use the commands below only for quick syntax and logic checks. Run them from the `Code` folder:

```
python3 "03. estimate_profit_bounds_fan_yang_2020.py" --max-tracts 10 --workers 1
python3 "04. compute_fixed_cost_fan_yang_2020.py" --n-tracts 500 --output-root /tmp/fy_test
python3 "06. simulate_baseline_portfolio_br_outcomes_2020.py" --max-draws 2 --max-markets 100
python3 "07. simulate_acp_portfolio_br_nash_counterfactual_2020.py" --max-draws 2 --max-markets 100
python3 "08. simulate_bead_portfolio_br_nash_counterfactual_2020.py" --max-draws 2 --max-markets 100
```

Use `-output-root` when smoke-testing script 04. Otherwise the test can overwrite full-sample fixed-cost outputs. The smoke-test modes for scripts 06–08 write to the real `data/` folder. Treat those files as temporary checks, not final results.

The full Fan–Yang profit-bound and fixed-cost steps (scripts 03–04) can take 1–2 hours on 70 854 tracts. Scripts 06–08 take roughly 30–90 minutes each with 50 draws.

5 Script-by-Script Summary

Script	Main task	Key outputs
00	Constructs upstream structural files: market, provider, product-space, demand-estimation, and entry-estimation inputs.	markets_2020.csv, product_space_2020.csv, demand_estimation_2020.csv, entry_estimation_2020.csv.
01	Constructs CSS broadband subscription shares used in the demand input.	demand_supply_2020.csv, css_logit_params_2020.csv, css_demand_documentation_2020.tex.
02	Estimates nested-logit demand and supply primitives.	nested_logit_parameters_2020.csv, nested_logit_markups_2020.csv, nested_logit_mc_ols_nofe_2020.csv.
03	Computes Fan–Yang expected variable-profit bounds using empirical profit-shock draws.	profit_bounds_fan_yang_2020.csv, potential_products_fan_yang_2020.csv.
04	Estimates seven fixed-cost parameters via moment inequalities; draws 50 FC vectors from the Andrews–Soares confidence region; writes comprehensive documentation PDF.	fixed_cost_simulation_draws_2020.csv, fixed_cost_fan_yang_2020_paper_table.csv, fixed_cost_fan_yang_2020_documentation.pdf.
05	Builds the ACS FPL eligibility file used by the ACP counterfactual.	data/acs_tract_2020_fpl.csv.
06	Solves baseline portfolio best-response equilibria for each draw; reports welfare, prices, HHI, level counts (products H/L, active firms, coverage, averages per tract), and market-share metrics.	baseline_portfolio_br_nash_draw_intervals_2020.csv, baseline_portfolio_br_nash_trimmed_bounds_2020.csv, baseline_portfolio_br_nash_documentation.pdf.
07	Simulates ACP consumer-price discounts ($10/$20/$30); same outputs as script 06 plus % change bounds vs baseline; loads FPL eligibility from ACS file.	acp_portfolio_br_nash_draw_intervals_2020.csv, acp_portfolio_br_nash_documentation.pdf.
08	Simulates BEAD as a proportional fixed-cost reduction (25/50/75%); same outputs as script 07.	bead_portfolio_br_nash_draw_intervals_2020.csv, bead_portfolio_br_nash_documentation.pdf.
09	Combines all draw-level results into CBA tables (NPV, BCR, net surplus) and figures; writes comprehensive documentation PDF.	cba_draw_level_2020.csv, cba_trimmed_bounds_2020.csv, cba_doc.pdf.

6 Data Sources

Table 1: Primary data sources

Dataset	Provider	Vintage	Link / Access
FCC Broadband Deployment Data (Form 477)	Federal Communications Commission	December 2020	https://www.fcc.gov/general/broadband-deployment-data-fcc-form-477
FCC Urban Rate Survey (URS)	Federal Communications Commission	2020	https://www.fcc.gov/economics-analytics/industry-analysis-division/urban-rate-survey-data-resources

Dataset	Provider	Vintage	Link / Access
FCC Consumer Survey on Services (CSS) / Broadband Subscription Shares	Federal Communications Commission	2020	https://www.fcc.gov/economics-analytics/industry-analysis-division/broadband-subscription-data
ACS 5-year Estimates — Demographic & Economic Characteristics	U.S. Census Bureau	2019 (5-yr)	https://data.census.gov (tables B01003, B19013, B25003, B15003, B01001)
ACS Table C17002 — Ratio of Income to Poverty Level	U.S. Census Bureau	2019 (5-yr)	https://data.census.gov/table/ACSDT5Y2019.C17002
ACS Table B28002 — Broadband / Internet Access	U.S. Census Bureau	2019 (5-yr)	https://data.census.gov/table/ACSDT5Y2019.B28002
Census TIGER/Line Shapefiles — Census Tracts	U.S. Census Bureau	2020	https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html

Notes on access. FCC Form 477 data can be downloaded from the FCC broadband map portal or the bulk download page. The URS workbook is distributed as an Excel file (`.xlsx`); if the system Python lacks Excel support, convert to CSV first. ACS tables can be retrieved via the Census API at https://api.census.gov/data with geography `tract` and year `2019`.

7 Raw Inputs and Environment

The raw files needed to rebuild the early-stage data are:

Script	Input	Role
00	`FCC477/form477_2020_december.csv.part`	Provider-by-block availability from FCC Form 477.
00	`FCC_shrs/tract_speed_category_shares_2020.csv`	Tract-level speed-category subscription shares.
00	`ACS/acs_tract_2019.csv`	Tract demographics and market-size controls.
00	`URS/ursdata2020.xlsx`	Residential broadband prices used for product-level price assignment.
01	`data/demand_estimation_2020.csv`	Observed product-tract rows produced by script 00.
05	`ACS/acs_tract_2020.csv`; Census ACS 2020 API, Table C17002	State list and FPL population counts for ACP eligibility shares.

Script 05 can use a `CENSUS_API_KEY` environment variable, but it does not require one. Script 00 reads the URS workbook as an Excel file; the Python environment therefore needs an Excel reader for `.xlsx` files.

Python environment. The scripts use `pandas`, `numpy`, `statsmodels`, `linearmodels`, `scipy`, and `matplotlib`. Documentation PDFs require either `tectonic` or `pdflatex`. The package was tested on Python 3.14 and requires Python 3.9 or later.

8 Required Inputs

If rebuilding the demand input from raw sources, run script 00 to construct the structural input files and script 01 to construct CSS broadband subscription shares. The current package then uses `data/finaldataset.csv`, the 2020 structural dataset read by script 02. Existing documentation records this file as a merge of data/entry_estimation_2020.csv and data/demand_supply_2020.csv, with the zero-share and cleanup filters applied before estimation.

After script 02 has run, the minimum downstream inputs are:

- data/nested_logit_parameters_2020.csv: preferred demand parameter file with alpha_price_100 and rho.
- data/nested_logit_markups_2020.csv: product-level demand and cost residuals used for Fan–Yang profit-shock draws.
- data/nested_logit_mc_ols_nofe_2020.csv: marginal-cost regression coefficients used for missing-cost imputation.
- profit_bounds_fan_yang_2020.csv: expected profit bounds used in fixed-cost estimation. This file is produced by script 03.
- fixed_cost_simulation_draws_2020.csv: 50 fixed-cost columns, fc_draw_001 through fc_draw_050. This file is produced by script 04.
- data/acs_tract_2020_fpl.csv: tract-level FPL eligibility shares used by the ACP counterfactual. This file is produced by script 05.
- data/potential_products_fan_yang_2020.csv: product primitives used by the baseline and policy simulations.

Script 07 (ACP) requires share_pop_below_200_fpl for each tract. If this column is absent from the draw file, the script loads it automatically from data/acs_tract_2020_fpl.csv (ACS 2019 Table C17002). Missing values are imputed by county median; any remaining gaps use the overall sample median.

9 Key Modeling Choices

Demand and Profit Bounds

Demand is estimated with a nested-logit model. Downstream simulations read the preferred specification from data/nested_logit_parameters_2020.csv; they do not use hard-coded demand parameters. For product j in tract-market m, with nest g, the estimated demand equation is

$$\log s_{jm} - \log s_{0m} = x'_{jm}\beta + \alpha_{100}\frac{p_{jm}}{100} + \rho \log s_{j|g,m} + \xi_{jm}.$$

The preferred values in the current package are approximately

$$\hat{\alpha}_{100} = -1.4907, \qquad \hat{\rho} = 0.8629.$$

On the supply side, script 02 recovers marginal costs from Bertrand–Nash price competition, using the estimated demand derivatives and observed prices.

The Fan–Yang objects are expected variable-profit bounds, not deterministic zero-shock profits. The run log records 100 empirical profit draws using paired demand and cost residuals. The relevant log is data/fan_yang_run_log_2020.txt.

Fixed-Cost Model

The fixed-cost model has seven parameters:

$$(\theta_S, \theta_M, \theta_L, \theta_H, \sigma_S, \sigma_M, \sigma_L).$$

The three θ parameters vary by market-size tertile, θ_H is the high-speed tier premium, and the three σ parameters are market-size-specific fixed-cost shock standard deviations.

Important settings in script 04 (changed from earlier versions):

- $\sigma_b \geq 0$ only — no lower-bound constraint other than strict positivity via the exp transform. The optimizer and stochastic search are both unconstrained on $\log \sigma_b$.
- **Confidence-set search**: 50 local chains × 50 steps on all available CPUs, followed by 100 expansion chains seeded from boundary points. Chains start from a grid of diverse σ values $\{10^{-10}, 100, 500, 1000, 5000, 10000\}$ combined with the optimizer's θ estimates, cycling across chains. This prevents degenerate starts when the optimizer converges to $\sigma \approx 0$.
- **Perturbation scale**: $\max(250, 0.2|\hat{\theta}_b|)$ for each θ parameter; 0.2 (log scale) for each σ. Expansion phase uses 5× these scales.
- **Seeds**: optimizer starts use 7 starting vectors ($\sigma \in \{250, 500, 1000, 2500, 5000, 10000\}$ plus data-based init); confidence-set seed `20240525`; simulation seed `20240524`.

The simulation uses 50 accepted parameter vectors from the confidence region. For each vector, fixed costs are drawn so that observed portfolios are rationalized as equilibria under the Fan–Yang bounds.

Baseline Portfolio Game

For each fixed-cost draw, the baseline script solves a tract-level portfolio game. Each provider chooses among feasible product portfolios: no product, low tier, high tier, or both tiers. Best responses are iterated from several starting configurations. Distinct fixed points that pass a firm-by-firm Nash check define the market-level equilibrium set. When multiple equilibria are found, the script reports lower and upper outcome bounds.

ACP Counterfactual

ACP is implemented as a targeted consumer-price discount:

$$p^c_{jt}(a) = \max\{p^g_{jt} - a, 0.01\}, \qquad a \in \{10, 20, 30\}.$$

The market is split into an eligible segment and an ineligible segment:

$$M^{elig}_t = M_t \times \texttt{share\_pop\_below\_200\_fpl}_t, \qquad M^{inelig}_t = M_t \times (1 - \texttt{share\_pop\_below\_200\_fpl}_t)\,.$$

Only the eligible segment receives the discount and faces $p^c_{jt}(a)$. The ineligible segment keeps the original consumer price p^g_{jt}. The gross provider price and marginal cost determine the provider margin. Fixed costs are unchanged. Government outlay equals the discount times eligible quantity.

BEAD Counterfactual

BEAD is modeled as a stylized fixed-cost subsidy:

$$FC^{cf}_{jt}(\tau) = (1 - \tau) FC_{jt}, \qquad \tau \in \{0.25, 0.50, 0.75\}.$$

This is a stylized version of BEAD's supply-side deployment incentive, not a literal simulation of all statutory targeting rules. The CBA step computes the fiscal cost as

$$G^{BEAD}(\tau) = \frac{\tau}{1 - \tau} \times PFC(\tau) \times 12,$$

where $PFC(\tau)$ is the counterfactual private fixed cost of active products.

10 Main Outcomes

Scripts 06–08 report draw-level lower and upper bounds for the metrics below. Script 09 adds CBA metrics. Reported intervals use $[Q_{2.5}(\underline{Y}^r), Q_{97.5}(\overline{Y}^r)]$ across 50 draws.

Metric group	Variables
Welfare	Consumer surplus, producer surplus ($B/month)
Prices & costs	Avg price, avg marginal cost ($/month)
Concentration	Market-size-weighted avg tract HHI (0–10,000)
Market share	Avg product market share $\bar{s}_{jt}$; overall take-up rate
Level counts	Total products offered; H products; L products; active firms (firm-market pairs); covered markets (tracts ≥ 1 product)
Avg per tract	Avg products/tract; avg H/tract; avg L/tract; avg firms/tract
Entry (changes)	New products H/L; new firm-markets; % entry rates vs baseline
ACP only	Government outlay; eligible/ineligible quantities; total surplus net of outlay

Scripts 07 and 08 also report a **percentage-change table** relative to baseline. For baseline interval $[a, b]$ and counterfactual interval $[c, d]$, the script computes two boundary-by-boundary changes and sorts them:

$$[\%\Delta^L, \%\Delta^U] = \left[\min\left(\frac{c-a}{a}, \frac{d-b}{b}\right),\ \max\left(\frac{c-a}{a}, \frac{d-b}{b}\right)\right] \times 100.$$

The lower baseline boundary a is paired with the lower counterfactual boundary c; the upper baseline boundary b is paired with the upper counterfactual boundary d. The two values are sorted so that $\%\Delta^L \leq \%\Delta^U$.

11 Cost-Benefit Analysis

Script 09 merges each policy result with the baseline result by `draw`. The interval arithmetic is conservative:

$$\underline{\Delta Y}^r = \underline{Y}^r_{cf} - \overline{Y}^r_{base}, \qquad \overline{\Delta Y}^r = \overline{Y}^r_{cf} - \underline{Y}^r_{base}.$$

For BEAD, monthly welfare changes are converted to annual flows and discounted over a 25-year horizon. Script 09 reports BEAD sensitivity at discount rates $r \in \{1\%, 3\%, 5\%, 7\%\}$, with $r = 3\%$ as the benchmark. For ACP, government outlays and welfare changes are annual recurring flows, so the main statistics are annual net surplus and benefit-cost ratios.

12 Final Deliverables

After the full pipeline runs, the main files to check are:

Object	Files
Structural dataset	`data/finaldataset.csv`; supporting files `data/markets_2020.csv`, `data/product_space_2020.csv`, `data/entry_estimation_2020.csv`, `data/demand_supply_2020.csv`.
Demand and supply	`data/nested_logit_parameters_2020.csv`, `data/nested_logit_markups_2020.csv`, `data/nested_logit_mc_ols_nofe_2020.csv`, `documentation/nest_demand_supply_doc.pdf`.
Profit bounds	`data/profit_bounds_fan_yang_2020.csv`, `data/potential_products_fan_yang_2020.csv`, `data/profit_bounds_quality_summary_2020.csv`, `graphic/fig_fan_yang_bounds_overview.pdf`.
Fixed costs	`data/fixed_cost_simulation_draws_2020.csv`, `data/fixed_cost_simulation_draws_2020_parameters.csv`, `data/fixed_cost_fan_yang_2020_paper_table.csv`, `documentation/fixed_cost_fan_yang_2020_documentation.pdf`.
Policy simulations	`data/baseline_portfolio_br_nash_draw_intervals_2020.csv`, `data/acp_portfolio_br_nash_draw_intervals_2020.csv`, `data/bead_portfolio_br_nash_draw_intervals_2020.csv`, with matching trimmed-bound files and documentation PDFs.
Cost-benefit analysis	`data/cba_draw_level_2020.csv`, `data/cba_trimmed_bounds_2020.csv`, `documentation/cba_doc.pdf`, `graphic/fig_cba_bcr.pdf`.

13 Verification Checklist

Before reporting final results, verify:

1. `data/finaldataset.csv` exists and, if rebuilt, reflects the documented merge of `entry_estimation_2020.csv` with `demand_supply_2020.csv`.
2. Script 01 writes `data/demand_supply_2020.csv` and `data/css_logit_params_2020.csv`.
3. The script 03 run log confirms empirical shock draws, not zero-shock profits.
4. Script 04 writes exactly 50 draw columns in `fixed_cost_simulation_draws_2020.csv`.
5. `fixed_cost_simulation_draws_2020_parameters.csv` reports `all_rows_best_response = True` for all 50 draws.
6. Script 04 confidence-set search yields at least 50 accepted points (check printed output: "Total accepted: N").
7. Scripts 06–08 all read the same `nested_logit_parameters_2020.csv` and the same `fixed_cost_simulation_draws_2020.c`
8. Draw-level outputs from scripts 06–08 contain the level-count columns: `offered_products_H_lower`, `active_firms_lower`, `avg_products_per_tract_lower`, etc. (required for % change table and CBA).
9. Script 07 prints no `ValueError` about `share_pop_below_200_fpl`; check that FPL imputation ran (county-median fallback message printed if any tract was missing).
10. Script 09 reads from `data/` and writes outputs to `data/` and `documentation/`. Confirm that `data/cba_draw_level_2020.csv` exists after run.
11. All documentation PDFs recompile without errors (check `documentation/*.log` for LaTeX errors).

14 Reproducibility Notes

Runtime. Script 04 takes roughly 1–2 hours on 70 854 tracts (parameter estimation + confidence-set search). Scripts 06–08 take 30–90 minutes each with 50 draws and 10 workers.

Draw consistency. Do not mix outputs from different fixed-cost draw files. The CBA merges on `draw`; draw r must correspond to the same fixed-cost realization in scripts 06, 07, and 08. When re-estimating fixed costs or changing the 50-draw selection, rerun scripts 06–09 in order.

Output files. In the current package, scripts write documentation as compiled PDFs in `documentation/` and figures in `graphic/`. Key documentation files:

- `fixed_cost_fan_yang_2020_documentation.pdf` — full replication details for script 04 (seeds, perturbation scales, weight-function cutoffs, sigma constraints).
- The baseline, ACP, and BEAD portfolio best-response documentation PDFs — per-draw tables, descriptive statistics, equilibrium diagnostics, and figures for scripts 06–08.
- `cba_doc.pdf` — full cost-benefit analysis (script 09).

External data dependency. Script 07 reads `data/acs_tract_2020_fpl.csv` for eligible-population shares. This file is produced by script 05 from ACS FPL inputs. If the file already ships with the package, script 05 only needs to be rerun when the ACS extract or FPL definition changes.

Python version. The pipeline requires Python 3.9+. Tested on Python 3.14. Avoid Python 3.7 or earlier (`from __future__ import annotations` is required for dataclass type hints).