

Package ‘rfcipDemand’

August 27, 2025

Type Package

Title Estimate Federal Crop Insurance Program Demand Models

Version 0.0.0.9000

Author Francis Tsiboe [aut, cre] (<<https://orcid.org/0000-0001-5984-1072>>)

Maintainer Francis Tsiboe <ftsiboe@hotmail.com>

Contributor -

Reviewer -

Creator Francis Tsiboe

Description Tools to construct county–crop–practice–plan–unit panels from the USDA RMA Summary of Business (SOBTPU) and related sources, and to estimate FCIP demand systems with two-way cluster-robust covariance. The pipeline standardizes coverage measures, merges price and instrument variables, adds rental-rate and price-index controls, reconciles county acreage (FSA/NASS), and produces diagnostics including robust first-stage F-tests. Methods align with the empirical design in “The crop insurance demand response to premium subsidies Evidence from U.S. Agriculture” (Food Policy, 2023, 119(3)).

License GPL-3 + file LICENSE

URL <https://github.com/you/rfcipDemand>

BugReports <https://github.com/you/rfcipDemand/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports rfcip, rfsa, data.table, systemfit, sandwich, doBy, car, plyr, usmap, stats, utils, tidyr

Remotes github::dylan-turner25/rfcip, github::UrbanInstitute/urbnmapr, github::dylan-turner25/rfsa, github::JanMarvin/nlsur

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

LazyData true

Cite-us If you find it useful, please consider starring the repository and citing the following studies
- Tsiboe, F. and Turner, D. (2025). “Incorporating buy-up price loss coverage into the United States farm safety net.” Applied Economic Perspectives and Policy.

- Tsiboe, F., et al. (2025). ``Risk reduction impacts of crop insurance in the United States." Applied Economic Perspectives and Policy.
- Gaku, S. and Tsiboe, F. (2024). Evaluation of alternative farm safety net program combination strategies. Agricultural Finance Review.

Contents

estimate_fcip_instruments	2
estimate_fcip_unloaded_rate	3
fcip_demand_data_controls	5
fcip_demand_data_dispatcher	6
fcip_demand_data_finalize	7
fcip_demand_data_prep_sob	7
fcip_demand_data_reconcile_acreage	9
fcip_demand_sys_coeff_table	10
fcip_demand_sys_effect	10
fcip_demand_sys_estimate	11
fcip_demand_sys_fit	11
fcip_demand_sys_level_prep	12
fcip_demand_sys_level_run	13
fcip_demand_sys_partial	13
fcip_demand_sys_restricted	14
fcip_demand_sys_tests	15
fcip_demand_sys_vcov	16
fixed_effect_model_data_prep	17
get_yu2018_instrument	18
panel_based_spatial_smoothing_estimator	19
Index	21

estimate_fcip_instruments
<i>Estimate FCIP Instrumental Variables (Unloaded Rates)</i>

Description

Uses historical FCIP rate data to build instrumented unloaded-rate variables following:

1. Tsiboe & Turner (2023), Econometric identification of crop insurance participation *Agricultural and Resource Economics Review*, 52(3):476-497. <https://doi.org/10.1017/age.2023.13>

Usage

```
estimate_fcip_instruments(year, statplan)
```

Arguments

<code>year</code>	Integer. The target crop year for which to construct instruments.
<code>statplan</code>	A <code>data.table</code> containing FCIP rate elements, including at least: <ul style="list-style-type: none"> <code>commodity_year</code> Year of the rate observation. <code>state_code, county_code</code> County identifiers. <code>commodity_code</code> Crop identifier. <code>insured_area, lcr, contiguous_state_code, contiguous_county_code</code> Fields required by <code>estimate_fcip_unloaded_rate()</code>.

Details

1. **Task list:** Identify all unique (state, county) pairs with data in the 2-21 years before year.
2. **Unloaded-rate calculation:** For each county in `task_list`, call `estimate_fcip_unloaded_rate()` on the same 2-21 year window to get tau. Errors return NULL so processing continues.
3. **Contiguous-county smoothing:**
 - Build a lookup table of contiguous counties (using `contiguous_county`).
 - For each contiguous group, compute the mean tau to get `tau_c`.
4. **Merge & fill:** Left-join the raw `adm` and `contiguous_adm`; replace any zero/NA/Inf tau with the group mean `tau_c` into `tau_sob`.
5. **Cleanup:** Drop helper columns (`tau`, `tau_c`), remove invalid rows, add `commodity_year`, and return the result.

Value

A `data.table` with one row per county-crop for the specified year, containing:

`state_code, county_code, commodity_code` Keys.

`tau_sob` Smoothed unloaded rate (uses contiguous-county means to fill zeros/NAs).

`commodity_year` The input year, repeated.

See Also

[estimate_fcip_unloaded_rate](#)

Other FCIP Demand Estimation: [fcip_demand_data_controls\(\)](#), [fcip_demand_data_dispatcher\(\)](#), [fcip_demand_data_finalize\(\)](#), [fcip_demand_data_prep_sob\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#), [get_yu2018_instrument\(\)](#)

estimate_fcip_unloaded_rate

Estimate FCIP Unloaded (County) Rates

Description

Computes the unloaded loss cost rates (tau) for counties based on the FCIC Rate Methodology Handbook (2009), pp. 65-70.

Usage

```
estimate_fcip_unloaded_rate(
  statplan,
  year = 2011,
  crop = NULL,
  state = NULL,
  county = NULL
)
```

Arguments

statplan	A data.table containing FCIP rate elements with at least the columns: state_code, county_code Identifiers for each county. contiguous_state_code, contiguous_county_code Mapping to county group. insured_area Total insured acres in the county. lcr Loss Cost Rate for each county. commodity_code Crop identifier.
year	Integer. Crop year for which rates are being estimated (currently not used but reserved for future subsetting).
crop	Optional vector of commodity codes to filter by crop.
state	Optional vector of state codes to restrict the analysis.
county	Optional vector of county codes to restrict the analysis.

Details

1. **Target data** is filtered to the selected state(s)/county(ies).
2. **Group data** finds contiguous-county groupings, unions them with the target.
3. Computes group-level statistics:

- c_{α} : mean insured acres
- c_u : mean LCR
- c_a : variance of LCR

1. Computes target county statistics:

- c_x : mean LCR
- c_v : variance of LCR
- c_{net_acre} : total insured acres

1. Applies the blending formula

$$\tau = Zx + (1 - Z)u, \quad Z = P/(P + K)$$

where $P = c_{net_acre}/c_{\alpha}$, $K = c_v/c_a$.

Value

A `data.frame` with columns:

state_code, county_code, commodity_code Keys identifying county and crop.
tau Estimated FCIP county unloaded rate.

References

FCIC Rate Methodology Handbook APH (2009), pp. 65-70. <https://legacy.rma.usda.gov/pubs/2008/ratemethodology.pdf>

fcip_demand_data_controls

Add recodes, prices, instruments, rental rates, and price index

Description

Adds commodity prices (ADM; projected price with fallback to harvest price), instruments (tau plus benchmark subsidy rates 65/75), state-level land rental rates, and NASS "index for price received". Also performs within-group price imputation to fill sparse county-year * crop-type-practice cells.

Usage

```
fcip_demand_data_controls(df)
```

Arguments

`df` A data.table produced by [fcip_demand_data_prep_sob\(\)](#).

Value

The same data.table with added columns: price, tau, subsidy_rate_65, subsidy_rate_75, rent, index_for_price_recived.

Data sources

- ADM price release: adm_extracts/fcip_commodity_price.rds
- Instrument release: reps/fcip_demand_instruments.rds (uses tau_adm, fallback tau_sob)
- Assumes in-memory tables: nass_state_rental_rates, nass_index_for_price_recived

Note

Column name index_for_price_recived follows the source spelling.

See Also

Other FCIP Demand Estimation: [estimate_fcip_instruments\(\)](#), [fcip_demand_data_dispatcher\(\)](#), [fcip_demand_data_finalize\(\)](#), [fcip_demand_data_prep_sob\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#), [get_yu2018_instrument\(\)](#)

fcip_demand_data_dispatcher

Build dataset to estimate Federal Crop Insurance Program (FCIP) demand (modular pipeline)

Description

End-to-end pipeline that: (1) prepares SOBTPU and coverage aggregates, (2) adds prices/instruments/rental rates/price index, (3) reconciles county acreage (FSA + NASS), and (4) finalizes bins/labels/pooling for demand estimation.

Usage

```
fcip_demand_data_dispatcher(
  study_years = 2001:(as.numeric(format(Sys.Date(), "%Y")) - 1),
  identifiers = c("commodity_year", FCIP_INSURANCE_POOL, "insurance_plan_code",
    "unit_structure_code")
)
```

Arguments

study_years	Integer vector of commodity years to include. Defaults to 2001:(as.numeric(format(Sys.Date(), "%Y")) - 1).
identifiers	Character vector of grouping keys that define the aggregation grain. Must be columns present in SOBTPU (e.g., "commodity_year", FCIP_INSURANCE_POOL, "insurance_plan_code", "unit_structure_code", and-if desired-additional keys like "commodity_code" or "practice_code"). Enrichment joins for re-codes are performed only when the required keys are included in identifiers

Details

Aligned with Asche, Bakkerman, & Li (2023), *Food Policy*, 119(3):102505 ([doi:10.1016/j.foodpol.2023.102505](https://doi.org/10.1016/j.foodpol.2023.102505)). Requires internet access to download release .rds assets and several in-memory lookup tables (see stage docs).

Value

A data.table ready for FCIP demand estimation.

See Also

[fcip_demand_data_prep_sob\(\)](#), [fcip_demand_data_controls\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#), [fcip_demand_data_finalize\(\)](#)

Other FCIP Demand Estimation: [estimate_fcip_instruments\(\)](#), [fcip_demand_data_controls\(\)](#), [fcip_demand_data_finalize\(\)](#), [fcip_demand_data_prep_sob\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#), [get_yu2018_instrument\(\)](#)

fcip_demand_data_finalize

Finalize FCIP demand dataset

Description

Applies required log-based validity filters, enforces crop support thresholds, creates Mundlak pooling identifiers, forms subsidy-share bins, attaches Farm Bill period labels, derives a simple before/after indicator, and adds state names/abbreviations.

Usage

```
fcip_demand_data_finalize(df)
```

Arguments

`df` A data.table from [fcip_demand_data_reconcile_acreage\(\)](#).

Details

- **Filters:** drops rows where logs of key variables are 0, NA, Inf, or -Inf.
- **Support thresholds:** keep crops with ≥ 30 obs per year for ≥ 10 years.
- **Mundlak pooling:** produces pool and a singleton flag per cross-sectional key.
- **Binning:** subsidy_bins in 0.02 steps from 0.40 to 0.80 (inclusive, with clamp).
- **Labels:** period_farmbill factor (pre-1980 ... 2018) and period_combo ("Before"/"After" 2012).

Value

Final data.table ready for estimation.

See Also

Other FCIP Demand Estimation: [estimate_fcip_instruments\(\)](#), [fcip_demand_data_controls\(\)](#), [fcip_demand_data_dispatcher\(\)](#), [fcip_demand_data_prep_sob\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#), [get_yu2018_instrument\(\)](#)

fcip_demand_data_prep_sob

Prep SOBTPU and compute coverage/financial aggregates for demand estimation

Description

Loads RMA Summary of Business (SOBTPU) data, filters valid observations, normalizes coverage levels, computes coverage summaries, and collapses financials at a chosen identifier grain. Optionally enriches with recode tables depending on which keys are present in identifiers.

Usage

```
fcip_demand_data_prep_sob(
  study_years = 2001:(as.numeric(format(Sys.Date(), "%Y")) - 1),
  identifiers = c("commodity_year", FCIP_INSURANCE_POOL, "insurance_plan_code",
    "unit_structure_code")
)
```

Arguments

<code>study_years</code>	Integer vector of commodity years to include. Defaults to <code>2001:(as.numeric(format(Sys.Date(), "%Y")) - 1)</code> .
<code>identifiers</code>	Character vector of grouping keys that define the aggregation grain. Must be columns present in SOBTPU (e.g., "commodity_year", FCIP_INSURANCE_POOL, "insurance_plan_code", "unit_structure_code", and-if desired-additional keys like "commodity_code" or "practice_code"). Enrichment joins for recodes are performed only when the required keys are included in identifiers (see details below).

Details

Normalization of coverage: values > 1 are treated as percentages and converted to proportions, snapped to a 0.05 grid, and clamped to 0.50:0.95.

Aggregation:

- Per-identifier coverage summaries: max, mean, mode (dominant), and weighted average (weights = `net_reporting_level_amount`).
- `potential_liability_amount` = `liability_amount` / `coverage_level_percent`
- `coverage_level_percent_aggregate` = `liability_amount` / `potential_liability_amount`

Optional enrichment (requires certain keys in identifiers):

- Commodity grouping (`commodity_year` + `commodity_code`)
- Practice recodes (`commodity_year` + `commodity_code` + `practice_code`)
- Insurance plan recodes (`commodity_year` + `insurance_plan_code`, filtered to `triger_level == "Individual"`)

Environment requirements (for optional enrichment):

- `fcip_recodes_commodity_groupings`, `fcip_recodes_practice`, `fcip_recodes_insurance_plan`.
- A function `calculate_mode()` used to compute the coverage mode.

Value

A `data.table` at the chosen identifier grain with coverage aggregates and the columns listed under **What this stage produces**.

What this stage produces

- Coverage metrics: `coverage_level_percent_{max, avg, dominant, wavg, aggregate}`
- Financial totals: `net_reporting_level_amount`, `liability_amount`, `total_premium_amount`, `subsidy_amount`
- Ratios: `premium_per_liability`, `subsidy_per_premium`

Data source

Downloads a released .rds: USFarmSafetyNetLab/sob/sobtpu_all.rds (GitHub Releases).

See Also

Other FCIP Demand Estimation: [estimate_fcip_instruments\(\)](#), [fcip_demand_data_controls\(\)](#), [fcip_demand_data_dispatcher\(\)](#), [fcip_demand_data_finalize\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#), [get_yu2018_instrument\(\)](#)

fcip_demand_data_reconcile_acreage

Reconcile county acreage from FSA and NASS

Description

Builds county-year planted acres from FSA (fsaCropAcreage joined via fsa_crop_linker) and merges NASS county series (planted, bearing, harvested). Sets county_acreage choosing the first non-missing in the order: FSA planted - NASS planted - NASS bearing - NASS harvested. Intermediate columns are dropped.

Usage

```
fcip_demand_data_reconcile_acreage(df)
```

Arguments

df A data.table from [fcip_demand_data_controls\(\)](#).

Value

The same data.table with a single county_acreage column and without nassSurvey_AREA_* or fsa_planted_acres intermediates.

Data sources

- Package data: fsaCropAcreage (loaded via data(fsaCropAcreage))
- Linker: fsa_crop_linker (columns: crop_cd_fsa, crop, crop_yr)
- Release download: nass_extracts/nass_production_data.rds

See Also

Other FCIP Demand Estimation: [estimate_fcip_instruments\(\)](#), [fcip_demand_data_controls\(\)](#), [fcip_demand_data_dispatcher\(\)](#), [fcip_demand_data_finalize\(\)](#), [fcip_demand_data_prep_sob\(\)](#), [get_yu2018_instrument\(\)](#)

fcip_demand_sys_coeff_table

Tidy coefficient table with cluster-robust SEs (from supplied VCOV)

Description

Builds a clean coefficient table for a `systemfit` model using a **user-supplied covariance matrix** (e.g., two-way clustered from `fcip_demand_sys_vcov()`). Estimates come from `coef(fit)`, standard errors from `diag(vcMat)`, then Z-scores and two-sided normal p-values are computed. The demand column is inferred from the equation prefix in the coefficient names:

- "Gamma_*" to "Gamma"
- "Theta_*", "Theta1_*", "Theta2_*", ... to "Theta" Otherwise the prefix itself is used.

Usage

```
fcip_demand_sys_coeff_table(fit, vcMat, p_digits = 5)
```

Arguments

<code>fit</code>	A fitted <code>systemfit</code> object.
<code>vcMat</code>	A covariance matrix conformable with <code>coef(fit)</code> . Row/column names are used to align; if missing, positional alignment is assumed.
<code>p_digits</code>	Integer; number of digits to keep for p-values (default 5).

Value

A `data.frame` with columns: `demand`, `coef`, `Estimate`, `StdError`, `Zvalue`, `Pvalue`.

fcip_demand_sys_effect

Delta-method "total protection response"

Description

Combines equation-specific effects into a single "Total" effect for each regressor in `c(fields$endogenous, fields$included)` using `car::deltaMethod` and a supplied covariance matrix.

Usage

```
fcip_demand_sys_effect(fit, vcMat, fields, data)
```

Arguments

<code>fit</code>	A fitted <code>systemfit</code> object (the structural system).
<code>vcMat</code>	Covariance matrix conformable with <code>coef(fit)</code> (e.g., from <code>fcip_demand_sys_vcov()</code>).
<code>fields</code>	Named list carrying model fields; must include <code>outcome</code> , <code>endogenous</code> , and <code>included</code> .
<code>data</code>	Estimation data used to check variable availability and build delta-method expressions.

Value

A data.frame with rows demand="Total" and columns: demand, coef, Estimate, StdError, Zvalue, Pvalue.

fcip_demand_sys_estimate

System estimator (modular wrapper; preserves original outputs)

Description

Runs: per-level prep -> partial/tilda -> systemfit -> clustered VCOV -> delta-method totals -> (optional) restricted NLSUR -> diagnostics -> bind rows.

Usage

```
fcip_demand_sys_estimate(model, data)
```

Arguments

model	List with elements: outcome, endogenous, included, excluded (opt), partial (opt), FE (logical), disag (string colname), optional restrict (logical), name (string).
data	Input data.frame/data.table with all referenced columns plus pool and commodity_year.

Value

A data.frame aggregating results across all disaggregation levels.

fcip_demand_sys_fit

Build systemfit formulas and estimate the system

Description

Constructs the list of structural equations (g) and instrument sets (h), then runs systemfit() using OLS (when no excluded instruments) or 3SLS-GMM (when excluded instruments are present).

Usage

```
fcip_demand_sys_fit(
  data,
  fields,
  tilda_included,
  tilda_endogenous,
  tilda_excluded
)
```

Arguments

<code>data</code>	Estimation <code>data.frame</code> / <code>data.table</code> containing the <code>tilda_*</code> and <code>instr_*</code> variables referenced by the formulas.
<code>fields</code>	Named list with at least <code>outcome</code> , <code>included</code> , <code>endogenous</code> , and optionally <code>excluded</code> .
<code>tilda_included</code>	Character vector of residualized included regressor names (e.g., <code>"tilda_x1"</code>).
<code>tilda_endogenous</code>	Character vector of residualized endogenous regressor names (e.g., <code>"tilda_z1"</code>).
<code>tilda_excluded</code>	Character vector of instrument names (e.g., <code>"instr_z1"</code>), or <code>NULL</code> when no excluded instruments are used.

Value

A list with elements:

<code>fit</code>	Fitted <code>systemfit</code> object.
<code>g</code>	List of structural formulas.
<code>h</code>	List of instrument formulas.

`fcip_demand_sys_level_prep`

Prepare data for a single level

Description

Filters to one disaggregation level, drops incomplete/invalid rows, removes constant partials, and optionally demeanes via a fixed-effects helper.

Usage

```
fcip_demand_sys_level_prep(data, fields, level)
```

Arguments

<code>data</code>	Estimation dataset that already contains all columns referenced by <code>fields</code> .
<code>fields</code>	Named list: <code>outcome</code> , <code>endogenous</code> , <code>included</code> , <code>excluded</code> (opt), <code>partial</code> (opt), <code>FE</code> (logical), <code>disag</code> (column name).
<code>level</code>	One element of the disaggregation levels.

Value

A list: `data` (prepped), `NFE` (number of FE), `partial` (possibly reduced).

fcip_demand_sys_level_run

Orchestrate one disaggregation level

Description

Runs the full pipeline for a single level: level prep -> partial/tilda creation -> systemfit -> two-way clustered VCOV -> delta-method totals -> optional restricted step -> diagnostics; then returns a tidy coefficient table with metadata.

Usage

```
fcip_demand_sys_level_run(base_data, fields, level)
```

Arguments

base_data	Full estimation dataset (before subsetting to the level).
fields	Named list carrying model fields (see <code>fcip_demand_estimation()</code>), including <code>disag</code> , <code>FE</code> , <code>outcome</code> , <code>endogenous</code> , <code>included</code> , <code>optional excluded</code> , <code>partial</code> , <code>restrict</code> , and <code>name</code> .
level	One value of <code>fields\$disag</code> to estimate for (e.g., a crop name).

Value

A `data.frame` with columns `demand`, `coef`, `Estimate`, `StdError`, `Zvalue`, `Pvalue` and meta-columns `model`, `endogenous`, `FE`, `name`, `disag`, `level`.

fcip_demand_sys_partial

Residualize ("partial out") and build tilded / instrument variables

Description

If excluded instruments exist, runs first-stage OLS for each endogenous variable e : $e \sim 1 + \text{partial} + \text{included} + \text{excluded}$, storing the fitted values as `instr_e`. If `partial` is non-empty, it then regresses $\text{instr_e} \sim 1 + \text{partial}$ and replaces $\text{instr_e} \leftarrow \text{instr_e} - \widehat{\text{fitted}(\text{instr_e} \sim \text{partial})}$ (i.e., removes the partial component; conceptually $\widehat{\text{instr}_e}(\text{partial})$ but expressed here without raw LaTeX macros).

Outcomes, included, and endogenous variables are residualized on `partial` to create `tilda_<var>` (or copied if `partial` is empty).

Usage

```
fcip_demand_sys_partial(data, fields, partial_override = NULL)
```

Arguments

<code>data</code>	A <code>data.frame</code> / <code>data.table</code> with referenced variables.
<code>fields</code>	List with: <code>outcome</code> , <code>endogenous</code> , <code>included</code> , <code>optional excluded</code> , <code>optional partial</code> .
<code>partial_override</code>	Optional character vector to override <code>fields\$partial</code> .

Value

List with `data`, `tilda_included`, `tilda_endogenous`, `tilda_excluded`.

`fcip_demand_sys_restricted`

Optional restricted NLSUR step (when `restrict = TRUE`)

Description

If enabled and feasible, estimates a nonlinear SUR with re-parameterized coefficients (negative exponents) using the optional **nlsur** package and appends "restricted_" rows to the results. Skips gracefully if **nlsur** is not installed or the step fails.

Usage

```
fcip_demand_sys_restricted(
  restrict,
  res,
  fit,
  data,
  outcome,
  tilda_endogenous,
  tilda_excluded,
  tilda_included
)
```

Arguments

<code>restrict</code>	Logical flag; when <code>TRUE</code> attempt the restricted step.
<code>res</code>	Coefficient table from the unrestricted system (used to check signs).
<code>fit</code>	Fitted <code>systemfit</code> object from the unrestricted system.
<code>data</code>	Estimation data used to fit the restricted NLSUR model.
<code>outcome</code>	Character vector of outcome equation names (length 2 expected).
<code>tilda_endogenous</code>	Character vector of endogenous regressors used in the tilded system (e.g., "tilda_z1").
<code>tilda_excluded</code>	Character vector of excluded instruments (e.g., "instr_z1").
<code>tilda_included</code>	Character vector of included regressors ("tilda_x1", ...).

Value

A `data.frame` with rows for `Gamma`, `Theta`, and `Total` labeled `restricted_*`, or an empty `data.frame` if skipped.

Note

This step uses `nlsur::nlsur()` if available; it is optional and should be listed under Suggests in DESCRIPTION.

`fcip_demand_sys_tests` *System diagnostics: two-way robust first-stage F (+ optional approx. J)*

Description

Produces diagnostics **without** re-running GMM:

- **FTest**: joint relevance of excluded instruments in each first stage, using the same two-way (pool by crop year) cluster-robust covariance via `fcip_demand_sys_vcov()` with `kind = "lm"`. Reports the **minimum** F across endogenous regressors.
- **JTest** (optional): an *approximate* over-identification test computed as the sum of per-equation Sargan statistics $J_k \approx n_k R_k^2$ from regressions of equation residuals on that equation's instrument set. This is a quick check (not the system Hansen J).

Usage

```
fcip_demand_sys_tests(g, h, data, fit, NFE, approx_j = FALSE)
```

Arguments

<code>g</code>	List of system equations (the same formulas passed to <code>systemfit</code>).
<code>h</code>	List of instrument formulas (the same formulas passed to <code>systemfit</code>).
<code>data</code>	Estimation <code>data.frame</code> / <code>data.table</code> containing all variables in <code>g/h</code> plus clustering columns <code>pool</code> and <code>crop_yr</code> .
<code>fit</code>	A fitted <code>systemfit</code> object (used for <code>N</code> and <code>residCov_*</code> extraction).
<code>NFE</code>	Integer: number of absorbed fixed effects (for reporting only).
<code>approx_j</code>	Logical, compute the approximate (non-robust) Sargan J as described above. Default <code>FALSE</code> (returns <code>NA</code> for JTest).

Value

A `data.frame` with rows: `N`, `NFE`, `residCov_11`, `residCov_22`, `residCov_12`, `JTest`, `FTest`.

fcip_demand_sys_vcov *Two-way cluster-robust covariance for FCIP demand models*

Description

Computes a Cameron-Gelbach-Miller two-way cluster-robust covariance matrix using inclusion-exclusion: $V = V_{pool} + V_{year} - V_{pool_year}$. Works for both `systemfit` (stacked system) and `lm` (first-stage).

Usage

```
fcip_demand_sys_vcov(
  object,
  data,
  kind = c("systemfit", "lm"),
  pool_col = "pool",
  year_col = "crop_yr",
  NFE = 0L,
  n_partial = 0L,
  n_eq = NULL
)
```

Arguments

<code>object</code>	Fitted model: either a <code>systemfit</code> or <code>lm</code> .
<code>data</code>	Estimation data containing pool and year identifiers.
<code>kind</code>	One of <code>c("systemfit", "lm")</code> . If omitted, auto-detected.
<code>pool_col</code>	Name of the pool/cluster id column in data (default "pool").
<code>year_col</code>	Name of the year/time id column in data (default "crop_yr").
<code>NFE</code>	Integer; number of absorbed fixed effects (for df rescaling).
<code>n_partial</code>	Integer; count of variables partialled out per equation.
<code>n_eq</code>	Integer; number of equations (<code>length(object\$eq)</code> for <code>systemfit</code> , 1 for <code>lm</code>). You can override if needed.

Details

Rescaling. Let n be the number of observations (stacked across equations for `systemfit`). With k_{old} the number of coefficients and $k_{new} = k_{old} + NFE + n_{partial} * n_{eq}$, the returned matrix is scaled by $(n - k_{old} - 1) / (n - k_{new} - 1)$.

Row alignment (lm). Rows used by `lm` are inferred from `rownames(model.matrix(object))`. If they cannot be mapped back to data, the first `nobs(object)` rows are used.

Value

Covariance matrix aligned with `coef(object)`.

fixed_effect_model_data_prep

Prepare and demean data for fixed-effects models

Description

This function

1. Filters to complete cases on the specified panel, time, weight, variables, and output
2. If output is NULL, creates a dummy output column filled with 1s
3. Drops any panel with only one observation
4. Computes within-panel means for the output + each variable in varlist (`_mean_i`)
5. Computes overall sample means for the same set of variables (`_mean`)
6. Replaces each variable in varlist by `value - within_panel_mean + overall_mean`

Usage

```
fixed_effect_model_data_prep(
  data,
  varlist,
  panel,
  time,
  wvar = NULL,
  output = NULL
)
```

Arguments

<code>data</code>	A <code>data.frame</code> or <code>data.table</code> containing the data.
<code>varlist</code>	Character vector of variable names to be demeaned.
<code>panel</code>	Character vector of column name(s) defining the panel identifier.
<code>time</code>	Character scalar name of the time variable.
<code>wvar</code>	Character scalar name of a variable to keep but <i>not</i> demean (optional, default NULL).
<code>output</code>	Character scalar name of an output variable whose means are computed but not altered; if NULL, a dummy column named "output" is created (optional, default NULL).

Value

A list with components

- **data**: a `data.table` containing
 - the original panel, time, wvar, varlist, and output columns
 - two mean columns for each of `c(output, varlist)`: `<name>_mean_i` (within-panel) and `<name>_mean` (overall)
- **NFE**: the number of panels with more than one observation

See Also

Other Estimators panel models: [panel_based_spatial_smoothing_estimator\(\)](#)

get_yu2018_instrument *Formulate & Merge National Subsidy Rate Instrument (Yu et al., 2018)*

Description

Downloads the historical Summary of Business RDS and computes national subsidy-rate instruments at specified coverage levels, following Yu et al. (2018).

Usage

```
get_yu2018_instrument(
  dt,
  delivery_systems = c("RBUP", "FBUP"),
  plan_codes = c(1:3, 90, 44, 25, 42),
  coverage_levels = c(65, 75)
)
```

Arguments

dt	sobcov
delivery_systems	Character vector. Delivery systems to include; default c("RBUP", "FBUP").
plan_codes	Integer vector. Insurance plan codes to include; default c(1:3, 90, 44, 25, 42).
coverage_levels	Numeric vector. Percent coverage levels to keep; default c(65, 75).

Value

A data.table with columns: commodity_year, subsidy_rate_65, subsidy_rate_75.

See Also

Other FCIP Demand Estimation: [estimate_fcip_instruments\(\)](#), [fcip_demand_data_controls\(\)](#), [fcip_demand_data_dispatcher\(\)](#), [fcip_demand_data_finalize\(\)](#), [fcip_demand_data_prep_sob\(\)](#), [fcip_demand_data_reconcile_acreage\(\)](#)

panel_based_spatial_smoothing_estimator

Panel-based spatial smoothing estimator

Description

This function

1. Constructs spatially-varying treatment interactions (one variable per spatial unit)
2. Applies within-panel/time fixed-effects demeaning to both outcome and interactions
3. Fits an OLS model by hand (`lm.fit`) to recover one coefficient per spatial unit

Usage

```
panel_based_spatial_smoothing_estimator(
  data,
  output,
  treatment,
  time,
  panel,
  spatialvar
)
```

Arguments

<code>data</code>	A <code>data.table</code> or <code>data.frame</code> containing panel data.
<code>output</code>	Name of the outcome variable (character scalar).
<code>treatment</code>	Name of the treatment variable whose spatial effects we estimate (character scalar).
<code>time</code>	Name of the time variable (character scalar).
<code>panel</code>	Name(s) of the panel identifier variable(s) (character vector).
<code>spatialvar</code>	Name of the spatial grouping variable (e.g. county FIPS; character scalar).

Details

Internally, we

1. Build `treatment_code = I(spatialvar==code) * treatment` for each spatial unit code.
2. Call `fixed_effect_model_data_prep()` to demean the outcome and all `treatment_code` variables.
3. Assemble the design matrix $X = [\text{output_mean_i}, \text{treatment_}]$ and response y .
4. Solve $\hat{\beta} = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{y}$ via `lm.fit`.
5. Return a row per spatial unit with its coefficient.

Value

A `data.table` with columns:

- `estimate`: the estimated spatial-unit coefficient
- `county_fips`: the spatial unit identifier (5-digit FIPS)
- `state_code`, `county_code`: parsed FIPS components

See Also

Other Estimators panel models: [fixed_effect_model_data_prep\(\)](#)

Index

* Estimators panel models

fixed_effect_model_data_prep, [17](#)
panel_based_spatial_smoothing_estimator,
[19](#)

* FCIP Demand Estimation

estimate_fcip_instruments, [2](#)
fcip_demand_data_controls, [5](#)
fcip_demand_data_dispatcher, [6](#)
fcip_demand_data_finalize, [7](#)
fcip_demand_data_prep_sob, [7](#)
fcip_demand_data_reconcile_acreage,
[9](#)
get_yu2018_instrument, [18](#)

* FCIP rating helpers

estimate_fcip_unloaded_rate, [3](#)

data.table, [4](#)

estimate_fcip_instruments, [2](#), [5–7](#), [9](#), [18](#)

estimate_fcip_unloaded_rate, [3](#), [3](#)

fcip_demand_data_controls, [3](#), [5](#), [6](#), [7](#), [9](#), [18](#)

fcip_demand_data_controls(), [6](#), [9](#)

fcip_demand_data_dispatcher, [3](#), [5](#), [6](#), [7](#), [9](#),
[18](#)

fcip_demand_data_finalize, [3](#), [5](#), [6](#), [7](#), [9](#), [18](#)

fcip_demand_data_finalize(), [6](#)

fcip_demand_data_prep_sob, [3](#), [5](#), [6](#), [7](#), [9](#),
[18](#)

fcip_demand_data_prep_sob(), [5](#), [6](#)

fcip_demand_data_reconcile_acreage, [3](#),
[5–7](#), [9](#), [9](#), [18](#)

fcip_demand_data_reconcile_acreage(),
[6](#), [7](#)

fcip_demand_sys_coeff_table, [10](#)

fcip_demand_sys_effect, [10](#)

fcip_demand_sys_estimate, [11](#)

fcip_demand_sys_fit, [11](#)

fcip_demand_sys_level_prep, [12](#)

fcip_demand_sys_level_run, [13](#)

fcip_demand_sys_partial, [13](#)

fcip_demand_sys_restricted, [14](#)

fcip_demand_sys_tests, [15](#)

fcip_demand_sys_vcov, [16](#)

fcip_demand_sys_vcov(), [15](#)

fixed_effect_model_data_prep, [17](#), [20](#)

get_yu2018_instrument, [3](#), [5–7](#), [9](#), [18](#)

panel_based_spatial_smoothing_estimator,
[18](#), [19](#)