

# Package ‘rfcipDemand’

January 28, 2026

**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](mailto:ftsiboe@hotmail.com)>

**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/ftsiboe/rfcipDemand>

**BugReports** <https://github.com/ftsiboe/rfcipDemand/issues>

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

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

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

**Suggests** knitr, rmarkdown, tibble, dplyr, purrr, readr, lavaan, piggyback, 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., & Turner, D. (2023). “The crop insurance demand response to premium subsidies - Evidence from U.S. Agriculture”. Food Policy, 119(3). doi.org/10.1016/j.foodpol.2023.102505  
- Tsiboe, F., & Turner, D. (2023). Econometric identification of crop insurance participation. Agricultural and Resource Economics Review. 52(3) 476-497. doi.org/10.1017/age.2023.13

## Contents

adjust_agent_outcomes_by_elasticity . . . . .	2
adjust_indemnity_liability_per_acre . . . . .	5
agCensusAcres . . . . .	6
agCensusBFR . . . . .	6
calibrate_fcip_demand_elasticities . . . . .	7
estimate_fcip_instruments . . . . .	8
fcip_contiguous_county . . . . .	9
fcip_demand_data_dispatcher . . . . .	9
fcip_demand_elasticities_lavaan . . . . .	10
fcip_demand_sys_coeff_table . . . . .	11
fcip_demand_sys_effect . . . . .	11
fcip_demand_sys_estimate . . . . .	12
fcip_demand_sys_fit . . . . .	13
fcip_demand_sys_partial . . . . .	14
fcip_demand_sys_prep . . . . .	15
fcip_demand_sys_run . . . . .	15
fcip_demand_sys_tests . . . . .	16
fcip_demand_sys_vcov . . . . .	16
fcip_recodes_commodity_groupings . . . . .	17
fcip_recodes_insurance_plan . . . . .	18
fcip_recodes_practice . . . . .	18
fcip_recodes_type . . . . .	19
fixed_effect_model_data_prep . . . . .	19
format_fcip_demand_table . . . . .	20
fsa_crop_linker . . . . .	21
get_yu2018_instrument . . . . .	21
global_variables . . . . .	22
nassAgPriceMonthlyIndex . . . . .	22
nassSurveyMYAprice . . . . .	23
nassSurveyPriceRecivedIndex . . . . .	23
nassSurveyRentalRates . . . . .	24
premium_subsidy_schedule . . . . .	24
prevented_planting_adm . . . . .	25
<b>Index</b>	<b>26</b>

---

adjust\_agent\_outcomes\_by\_elasticity

*Adjust FCIP outcomes by price elasticities*

---

## Description

Simulates changes in Federal Crop Insurance Program (FCIP) outcomes- coverage level, insured acres, liability, premium, subsidy, and indemnity- under an alternative premium-per-liability using elasticities of insured acres and/or coverage level.

**Usage**

```

adjust_agent_outcomes_by_elasticity(
  alternate_premium_per_liability,
  insured_acres_elasticity,
  coverage_level_elasticity,
  baseline_coverage_type,
  baseline_coverage_level,
  baseline_insured_acres,
  baseline_liability_per_acre,
  baseline_premium_per_liability,
  baseline_subsidy_per_premium,
  baseline_indemnity_per_acre,
  baseline_subsidy_percent,
  baseline_rate_differential,
  final_revenue_per_acre,
  assumption = 0,
  premium_subsidy_schedule = NULL,
  rate_differential_schedule = NULL
)

```

**Arguments**

`alternate_premium_per_liability`  
Numeric. Alternative premium per dollar of liability.

`insured_acres_elasticity`  
Numeric. Elasticity of insured acres w.r.t. price (percent basis).

`coverage_level_elasticity`  
Numeric. Elasticity of coverage level w.r.t. price (percent basis).

`baseline_coverage_type`  
Character. Baseline coverage type code ("A" or "C").

`baseline_coverage_level`  
Numeric in (0,1]. Baseline coverage level share.

`baseline_insured_acres`  
Numeric. Baseline insured acres.

`baseline_liability_per_acre`  
Numeric. Baseline liability per acre.

`baseline_premium_per_liability`  
Numeric. Baseline premium per dollar of liability.

`baseline_subsidy_per_premium`  
Numeric in (0,1). Subsidy share of total premium.

`baseline_indemnity_per_acre`  
Numeric. Baseline indemnity per acre.

`baseline_subsidy_percent`  
Numeric. Baseline subsidy percent.

`baseline_rate_differential`  
Numeric. Baseline rate differential factor.

`final_revenue_per_acre`  
Numeric. Revenue per acre (used for indemnity adjustment).

`assumption`  
Integer (0,1,2,3). Scenario selector (see above). Default 0.

premium\_subsidy\_schedule

Optional numeric vector of length 8 corresponding to coverage levels 0.50, 0.55, ..., 0.85. Multiplicative factors applied to baseline\_subsidy\_per\_premium at the scenario coverage. Defaults to 1's.

rate\_differential\_schedule

Optional numeric vector of length 8 corresponding to coverage levels 0.50, 0.55, ..., 0.85. Multiplicative factors applied to alternate\_premium\_per\_liability at the scenario coverage. Defaults to 1's.

## Details

### Assumptions (set via assumption):

- 0: Fixed demand - both coverage and acres remain at baseline.
- 1: Acres respond to price (elasticity gamma); coverage fixed.
- 2: Coverage responds to price (elasticity tetha); acres fixed.
- 3: Both acres and coverage respond.

### Price response:

Price response is applied using the percent change in price:

$$\% \Delta p = 100 \times \left( \frac{\text{alt}}{\text{base}} - 1 \right),$$

and for any baseline quantity  $q$  with elasticity  $e_q$ :

$$q_{\text{new}} = q_{\text{base}} \left[ 1 + \left( e_q \times \frac{\% \Delta p}{100} \right) \right].$$

### Coverage adjustment:

Coverage is rounded to the nearest 0.05 grid and truncated to the range (0, 0.85). Values less than 0.50 are set to 0 (i.e., no coverage). Coverage type code is set to "A" for buy-up coverage and "C" for CAT coverage. For CAT policies, scenario coverage levels at 0.50/0.55 are adjusted to reflect CAT rules.

Per-acre liability and indemnity are updated using [adjust\\_indemnity\\_liability\\_per\\_acre](#).

### Optional schedules:

Coverage-specific scaling is possible via:

- premium\_subsidy\_schedule: length-8 numeric for coverage levels 0.50, 0.55, ..., 0.85. Multiplies the baseline\_subsidy\_per\_premium at the scenario coverage.
- rate\_differential\_schedule: length-8 numeric for the same grid. Multiplies the alternate\_premium\_per\_liability at the scenario coverage.

### Missing or zero inputs:

If either insured\_acres or coverage\_level\_percent is NA or 0 after adjustment, both are reset to 0 and all dollar outcomes are set to 0.

Elasticities are applied multiplicatively to baseline quantities. Coverage levels are snapped to the 0.05 grid and truncated. When schedules are supplied, subsidy shares and premium rates are rescaled relative to the baseline coverage before applying to the scenario coverage.

**Value**

A list with elements:

coverage\_type\_code Scenario coverage type ("A" or "C").  
 coverage\_level\_percent Scenario coverage level share.  
 insured\_acres Scenario insured acres.  
 liability\_amount Total liability = acres \* adjusted liability per acre.  
 total\_premium\_amount Total premium = liability \* alt premium/liability.  
 subsidy\_amount Subsidy = total premium \* subsidy share.  
 indemnity\_amount Total indemnity = acres \* adjusted indemnity per acre.  
 price\_change\_pct Percent price change used for elasticity adjustments.

---

adjust\_indemnity\_liability\_per\_acre

*Adjust liability and indemnity per acre at an alternative coverage level*

---

**Description**

Compute adjusted liability per acre and adjusted indemnity per acre when moving from a baseline coverage level to a new level. The method floors production-to-count at 0 and resolves five mutually exclusive cases via `data.table::fcase()`.

**Usage**

```
adjust_indemnity_liability_per_acre(
  coverage_level_percent,
  final_revenue_per_acre,
  baseline_coverage_level,
  baseline_liability_per_acre,
  baseline_indemnity_per_acre
)
```

**Arguments**

coverage\_level\_percent  
 Numeric vector; alternative coverage level (proportion in (0,1)).

final\_revenue\_per\_acre  
 Numeric vector; revenue per acre at harvest (used when baseline had no indemnity and baseline coverage is lower).

baseline\_coverage\_level  
 Numeric vector; baseline coverage level (proportion in (0,1)).

baseline\_liability\_per\_acre  
 Numeric vector; baseline liability per acre.

baseline\_indemnity\_per\_acre  
 Numeric vector; baseline indemnity per acre.

**Details**

- `production_to_count = pmax(baseline_liability_per_acre - baseline_indemnity_per_acre, 0)`
- `adj_Liability_per_acre = (baseline_liability_per_acre / baseline_coverage_level) * coverage_level_percent`
- Indemnity cases (I1-I5) selected with `data.table::fcase()`.

**Value**

A list with:

`adj_Liability_per_acre` Adjusted liability per acre.

`adj_Indemnity_per_acre` Adjusted indemnity per acre.

---

<code>agCensusAcres</code>	<i>agCensusAcres</i>
----------------------------	----------------------

---

**Description**

A combined dataset for `agCensusAcres`

**Usage**

```
data(agCensusAcres)
```

**Format**

A data frame with 15369 rows and 15 columns covering Inf–Inf.

**Source**

USDA NASS Quick Stats

---

<code>agCensusBFR</code>	<i>agCensusBFR</i>
--------------------------	--------------------

---

**Description**

A combined dataset for `agCensusBFR`

**Usage**

```
data(agCensusBFR)
```

**Format**

A data frame with 255 rows and 16 columns covering Inf–Inf.

**Source**

USDA NASS Quick Stats

---

calibrate\_fcip\_demand\_elasticities

*Calibrate FCIP demand elasticities*


---

## Description

Fits a 2-equation FCIP demand system over an estimation window ending in `calibration_year` and returns capped elasticities  $(-2, 0)$  by disaggregation level using `fcip_demand_sys_estimate()`

## Usage

```
calibrate_fcip_demand_elasticities(
  calibration_year,
  estimation_window,
  data,
  drawn_pools = NULL,
  disaggregate = NULL
)
```

## Arguments

<code>calibration_year</code>	Integer. Last year of the estimation window.
<code>estimation_window</code>	Integer ( $\geq 1$ ). Number of years ending at <code>calibration_year</code> .
<code>data</code>	<code>data.table</code> . Input panel. Must include: <code>commodity_year</code> , <code>commodity_code</code> , <code>drawID</code> , <code>pool</code> , <code>insurance_plan_code</code> , <code>price</code> , <code>net_reporting_level_amount</code> , <code>coverage_level_percent_aggregate</code> , <code>rent</code> , <code>county_acreage</code> , <code>total_premium_amount</code> , <code>subsidy_amount</code> , <code>liability_amount</code> , <code>tau</code> , <code>subsidy_rate_65</code> , <code>subsidy_rate_75</code> .
<code>drawn_pools</code>	Optional <code>data.table</code> used to filter data via an inner join on intersecting columns.
<code>disaggregate</code>	Optional character vector of grouping variables passed to the estimator (e.g., <code>"commodity_code"</code> or <code>c("state", "commodity_code")</code> ).

## Details

Internally, the function (i) restricts data to the estimation years, (ii) collapses insurance plan codes and sets `price := 1` for non-RP crops, (iii) constructs log variables, trend and year dummies, and (iv) estimates the system with rate endogenous and  $\tau_0$  as the excluded instrument. Estimates are truncated to  $(-2, 0)$  and returned in wide form.

## Value

`data.table` with columns: `disag`, `level`, `gamma_elasticity`, `theta_elasticity`.

---

estimate\_fcip\_instruments

*Estimate FCIP Instrumental Variables (Unloaded Rates)*


---

## 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

year	Integer. The target crop year for which to construct instruments.
statplan	A data.table containing FCIP rate elements, including at least: <b>commodity_year</b> Year of the rate observation. <b>state_code, county_code</b> County identifiers. <b>commodity_code</b> Crop identifier. <b>insured_area, lcr, contiguous_state_code, contiguous_county_code</b> Fields required by estimate_fcip_unloaded_rate().

## 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

Other FCIP instruments: [get\\_yu2018\\_instrument\(\)](#)

---

```
fcip_contiguous_county
      fcip_contiguous_county
```

---

**Description**

A combined dataset for fcip\_contiguous\_county

**Usage**

```
data(fcip_contiguous_county)
```

**Format**

A data frame with 24307 rows and 12 columns covering Inf–Inf.

**Source**

USDA-RMA, Actuarial Data Master - A0123

---

```
fcip_demand_data_dispatcher
      Build dataset to estimate Federal Crop Insurance Program (FCIP) de-
      mand (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.

---

fcip\_demand\_elasticities\_lavaan

*Calibrate FCIP demand elasticities via IV-SEM (lavaan)*

---

**Description**

Fits a just-identified IV-style SEM where instr\_rate instruments tilda\_rate, and tilda\_rate enters two outcome equations (tilda\_ghamma, tilda\_theta). Endogeneity is encoded via residual correlations between tilda\_rate and each outcome. The instrument is excluded from the outcome disturbances.

**Usage**

```
fcip_demand_elasticities_lavaan(
  data,
  estimator = c("ML", "MLR"),
  missing = c("fiml", "listwise")
)
```

**Arguments**

data	data.frame with cols: instr_rate, tilda_rate, tilda_gamma, tilda_theta
estimator	"ML" or "MLR" (default "MLR" = robust SE/tests)
missing	"fiml" or "listwise" (default "fiml")

**Details**

Constraints imposed:

1.  $b_1 < 0$ , 2)  $b_2 < 0$ , 3)  $b_1 + b_2 + b_1*b_2 < 0$

**Value**

A data.table of parameter estimates and logical convergence flag

---

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\_\*" 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**

Estimates a two-equation system with an endogenous regressor across disaggregation levels. The pipeline is: (1) per-level sample selection (min n per commodity\_year), (2) partialling-out / tilda transforms, (3) system estimation (e.g., IV/3SLS via internal helpers), (4) clustered variance estimation, (5) delta-method totals, (6) optional constrained re-estimation of elasticities (lavaan), (7) diagnostics, and (8) row-binding results across levels.

**Usage**

```
fcip_demand_sys_estimate(model, data, constrained_elasticities = FALSE)
```

**Arguments**

model	<p>List specifying the system; required elements are:</p> <ul style="list-style-type: none"> <li>• outcome (character(2)): names of the two outcomes in data.</li> <li>• endogenous (character(1)): endogenous regressor name.</li> <li>• included (character): included (exogenous) regressors.</li> <li>• disag (character(1) or NULL): disaggregation key column in data.</li> <li>• FE (logical): include fixed effects in the internal run (handled by helpers).</li> </ul> <p>Optional elements:</p> <ul style="list-style-type: none"> <li>• excluded (character or NULL): excluded instruments.</li> <li>• partial (character or NULL): variables to partial out (tilda).</li> <li>• restrict (logical): pass-through to internal restricted estimation.</li> <li>• name (character(1)): label carried to the output.</li> </ul>
data	data.frame/data.table containing all referenced columns in model plus pool and commodity_year. Columns in model\$outcome, model\$endogenous, model\$included, and (if used) model\$excluded and model\$partial must exist in data.
constrained_elasticities	<p>Logical (default FALSE). If TRUE, re-estimate the elasticities via a constrained SEM (lavaan) and, where applicable, replace positive elasticity estimates with constrained ones. See <b>Constrained elasticities (optional)</b>.</p>

**Details****Inputs and preprocessing**

- model\$outcome must be a character vector of length 2 giving the two outcome column names in data. Internally these are mapped to gamma and theta for estimation convenience.
- model\$disag is the name of the disaggregation key. If NULL, a dummy "full\_sample" key is created.

- The disaggregation key is coerced to character.
- For each level of `model$disag`, the function keeps only levels that have **at least 30 observations per commodity\_year** (computed via `doBy::summaryBy`). Levels failing this threshold are dropped.

**Per-level estimation** For each retained level, the function calls internal helpers (`fcip_demand_sys_run`, etc.) to (i) partially out controls if requested, and (ii) estimate the system with clustered VCOV and delta-method totals. Errors at the level are caught and the level is skipped (no hard stop).

**Constrained elasticities (optional)** When `constrained = TRUE`, elasticities on the endogenous regressor are re-estimated per level via a lavaan SEM with sign restrictions (internal helper `fcip_demand_elasticities_1` using `estimator = "MLR"`, `missing = "listwise"`). The constrained estimates replace any positive system estimates for the elasticities; a logical flag `constrained` marks levels where a replacement occurred. The returned columns are reduced to `disag`, `level`, `demand`, `constrained`, and `Estimate`.

#### Returned shape

- If `constrained_elasticities = FALSE` (default), returns the full per-level system output from `fcip_demand_sys_run` with additional columns: `disag`, `level`, and rounded `Zvalue`, `Pvalue`.
- If `constrained = TRUE`, returns a compact table with `disag`, `level`, `demand`, `constrained`, `Estimate` after merging the constrained elasticities.

#### Value

A `data.table` aggregating results across all disaggregation levels. The column set depends on `constrained` (see **Returned shape**).

---

<code>fcip_demand_sys_fit</code>	<i>Build systemfit formulas and estimate the system</i>
----------------------------------	---

---

#### 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> .

tilda\_included Character vector of residualized included regressor names (e.g., "tilda\_x1").

tilda\_endogenous Character vector of residualized endogenous regressor names (e.g., "tilda\_z1").

tilda\_excluded Character vector of instrument names (e.g., "instr\_z1"), or NULL when no excluded instruments are used.

### Value

A list with elements:

fit Fitted systemfit object.

g List of structural formulas.

h List of instrument formulas.

---

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  $\text{instr}_e \leftarrow \text{instr}_e - \text{fitted}(\text{instr}_e \sim \text{partial})$  (i.e., removes the partial component; conceptually  $\widehat{\text{instr}_e}(\text{partial})$ ).

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

data A data.frame/data.table with referenced variables.

fields List with: outcome, endogenous, included, optional excluded, optional partial.

partial\_override Optional character vector to override `fields$partial`.

### Details

Uses defensive checks so absent columns are ignored (with a warning) rather than erroring. If `partial` is empty, residualization is a no-op and `tilda_*` simply copy the originals. Formulas are constructed via `stats::reformulate()` to avoid paste/quoting pitfalls.

### Value

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

---

fcip\_demand\_sys\_prep    *Prepare data for analysis*


---

**Description**

Drops incomplete/invalid rows, removes constant partials, and optionally demeanes via a fixed-effects helper.

**Usage**

```
fcip_demand_sys_prep(data, fields)
```

**Arguments**

data	Estimation dataset that already contains all columns referenced by fields.
fields	Named list: outcome, endogenous, included, excluded (opt), partial (opt), FE (logical), disag (column name).

**Value**

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

---

fcip\_demand\_sys\_run    *Orchestrate analysis*


---

**Description**

Runs the full pipeline: 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_run(data, fields)
```

**Arguments**

data	Estimation dataset
fields	Named list carrying model fields (see <code>fcip_demand_estimation()</code> ), including disag, FE, outcome, endogenous, included, optional excluded, partial, restrict, and name.

**Value**

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

---

fcip_demand_sys_tests	<i>System diagnostics: two-way robust first-stage F (+ optional approx. J)</i>
-----------------------	--

---

## 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 data.frame/data.table containing all variables in g/h plus clustering columns <code>pool</code> and <code>crop_yr</code> .
<code>fit</code>	A fitted <code>systemfit</code> object (used for N 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 FALSE (returns NA for JTest).

## Value

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

---

fcip_demand_sys_vcov	<i>Two-way cluster-robust covariance for FCIP demand models</i>
----------------------	---

---

## 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 = "commodity_year",
  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 <code>"pool"</code> ).
<code>year_col</code>	Name of the year/time id column in data (default <code>"crop_yr"</code> ).
<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)`.

---

```
fcip_recodes_commodity_groupings
      fcip_recodes_commodity_groupings
```

---

**Description**

A combined dataset for `fcip_recodes_commodity_groupings`

**Usage**

```
data(fcip_recodes_commodity_groupings)
```

**Format**

A data frame with 3617 rows and 10 columns covering 1997-2026.

**Source**

USDA-RMA, Actuarial Data Master - A00400 and A00420 supplemented data from legacy ADM files

---

```
fcip_recodes_insurance_plan  
  fcip_recodes_insurance_plan
```

---

**Description**

A combined dataset for fcip\_recodes\_insurance\_plan

**Usage**

```
data(fcip_recodes_insurance_plan)
```

**Format**

A data frame with 802 rows and 10 columns covering 1989-2026.

**Source**

USDA-RMA, Actuarial Data Master - A00460 supplemented data from legacy ADM files

---

```
fcip_recodes_practice  fcip_recodes_practice
```

---

**Description**

A combined dataset for fcip\_recodes\_practice

**Usage**

```
data(fcip_recodes_practice)
```

**Format**

A data frame with 29848 rows and 8 columns covering 1997-2026.

**Source**

USDA-RMA, Actuarial Data Master - A00510 supplemented data from legacy ADM files

---

fcip_recodes_type	<i>fcip_recodes_type</i>
-------------------	--------------------------

---

**Description**

A combined dataset for fcip\_recodes\_type

**Usage**

```
data(fcip_recodes_type)
```

**Format**

A data frame with 234220 rows and 7 columns covering 1999-2026.

**Source**

Generated internally, using harmonize\_crop\_type\_codes()

---

fixed_effect_model_data_prep	<i>Prepare and demean data for fixed-effects models</i>
------------------------------	---

---

**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**

data	A data.frame or data.table containing the data.
varlist	Character vector of variable names to be demeaned.
panel	Character vector of column name(s) defining the panel identifier.
time	Character scalar name of the time variable.
wvar	Character scalar name of a variable to keep but <i>not</i> demean (optional, default NULL).
output	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

---

format\_fcip\_demand\_table

*Table: Crop Insurance Demand System for US Federal Crop Insurance Pools (2001/22)*

---

**Description**

Build a two-column, GitHub-safe panel table summarizing a crop insurance demand system. The table is organized into panels for coverage level (Theta), insured acres (Gamma), total protection response, a covariance matrix block, and additional statistics. Coefficients are formatted as estimate (std. error) with significance stars.

**Usage**

```
format_fcip_demand_table(df, var_labels)
```

**Arguments**

df	<p>A data frame containing the results with columns:</p> <ul style="list-style-type: none"> <li>• demand (chr): panel identifier; expected values include "Theta", "Gamma", and "Total".</li> <li>• coef (chr): raw coefficient/row labels (e.g., "tilda_rate", "residCov_11", "N").</li> <li>• Estimate (dbl): point estimates.</li> <li>• StdError (dbl): standard errors (may be NA for scalars).</li> <li>• Pvalue (dbl): p-values used to add significance stars.</li> </ul>
var_labels	A named character vector mapping raw names to display labels,

**Details**

Designed for README/output knitted as `github_document`; use with `knitr::kable(..., format = "pipe")` to avoid HTML-only features.

**Value**

A tibble with two columns, `Variables` and `Estimates`, where panel headers have empty `Estimates` to enable bolding (if rendered in HTML) and coefficients are formatted as `"estimate*** (se)"`.

---

fsa_crop_linker	<i>Simulator Helper Datasets</i>
-----------------	----------------------------------

---

**Description**

A combined dataset for `fsa_crop_linker`

**Usage**

```
data(fsa_crop_linker)
```

**Format**

A data frame with 8778 rows and 8 columns covering Inf–Inf.

**Source**

Internal innovation

---

get_yu2018_instrument	<i>Formulate &amp; Merge National Subsidy Rate Instrument (Yu et al., 2018)</i>
-----------------------	---

---

**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 instruments: [estimate\\_fcip\\_instruments\(\)](#)

---

global_variables	<i>fsa_crop_linker</i>
------------------	------------------------

---

Description

A combined dataset for global\_variables

Usage

global\_variables

Format

A data frame with 8594 rows and 8 columns covering Inf–Inf.

Source

Internal innovation

---

nassAgPriceMonthlyIndex	<i>nassAgPriceMonthlyIndex</i>
-------------------------	--------------------------------

---

Description

A combined dataset for nassAgPriceMonthlyIndex

Usage

data(nassAgPriceMonthlyIndex)

**Format**

A data frame with 2799 rows and 8 columns covering Inf–Inf.

**Source**

USDA NASS: [https://www.nass.usda.gov/Charts\\_and\\_Maps/graphics/data](https://www.nass.usda.gov/Charts_and_Maps/graphics/data)

---

nassSurveyMYAprice	<i>nassSurveyMYAprice</i>
--------------------	---------------------------

---

**Description**

A combined dataset for nassSurveyMYAprice

**Usage**

```
data(nassSurveyMYAprice)
```

**Format**

A data frame with 33078 rows and 7 columns covering 1866-2024.

**Source**

USDA NASS Quick Stats

---

nassSurveyPriceRecivedIndex	<i>nassSurveyPriceRecivedIndex</i>
-----------------------------	------------------------------------

---

**Description**

A combined dataset for nassSurveyPriceRecivedIndex

**Usage**

```
data(nassSurveyPriceRecivedIndex)
```

**Format**

A data frame with 35 rows and 3 columns covering 1990-2024.

**Source**

USDA NASS Quick Stats

---

nassSurveyRentalRates	<i>nassSurveyRentalRates</i>
-----------------------	------------------------------

---

**Description**

A combined dataset for nassSurveyRentalRates

**Usage**

```
data(nassSurveyRentalRates)
```

**Format**

A data frame with 1792 rows and 5 columns covering 1994-2025.

**Source**

Output from get\_state\_rental\_rates() function

---

premium_subsidy_schedule	<i>premium_subsidy_schedule</i>
--------------------------	---------------------------------

---

**Description**

A combined dataset for premium\_subsidy\_schedule

**Usage**

```
data(premium_subsidy_schedule)
```

**Format**

A data frame with 7951 rows and 7 columns covering 2001-2026.

**Source**

USDA-RMA, Actuarial Data Master supplemented data from legacy ADM files

---

prevented_planting_adm
<i>prevented_planting_adm</i>

---

**Description**

A combined dataset for prevented\_planting\_adm

**Usage**

`data(prevented_planting_adm)`

**Format**

A data frame with 252605 rows and 10 columns covering 2011-2026.

**Source**

USDA-RMA, Actuarial Data Master supplemented data from legacy ADM files

# Index

- \* **Estimation Data**
  - [fcip\\_demand\\_data\\_dispatcher, 9](#)
- \* **Estimators panel models**
  - [fixed\\_effect\\_model\\_data\\_prep, 19](#)
- \* **FCIP instruments**
  - [estimate\\_fcip\\_instruments, 8](#)
  - [get\\_yu2018\\_instrument, 21](#)
- \* **datasets**
  - [agCensusAcres, 6](#)
  - [agCensusBFR, 6](#)
  - [fcip\\_contiguous\\_county, 9](#)
  - [fcip\\_recodes\\_commodity\\_groupings, 17](#)
  - [fcip\\_recodes\\_insurance\\_plan, 18](#)
  - [fcip\\_recodes\\_practice, 18](#)
  - [fcip\\_recodes\\_type, 19](#)
  - [fixed\\_effect\\_model\\_data\\_prep, 19](#)
  - [format\\_fcip\\_demand\\_table, 20](#)
  - [fsa\\_crop\\_linker, 21](#)
  - [get\\_yu2018\\_instrument, 8, 21](#)
  - [global\\_variables, 22](#)
  - [nassAgPriceMonthlyIndex, 22](#)
  - [nassSurveyMYAprice, 23](#)
  - [nassSurveyPriceRecivedIndex, 23](#)
  - [nassSurveyRentalRates, 24](#)
  - [premium\\_subsidy\\_schedule, 24](#)
  - [prevented\\_planting\\_adm, 25](#)
- [adjust\\_agent\\_outcomes\\_by\\_elasticity, 2](#)
- [adjust\\_indemnity\\_liability\\_per\\_acre, 4, 5](#)
- [agCensusAcres, 6](#)
- [agCensusBFR, 6](#)
- [calibrate\\_fcip\\_demand\\_elasticities, 7](#)
- [estimate\\_fcip\\_instruments, 8, 22](#)
- [fcip\\_contiguous\\_county, 9](#)
- [fcip\\_demand\\_data\\_dispatcher, 9](#)
- [fcip\\_demand\\_elasticities\\_lavaan, 10](#)
- [fcip\\_demand\\_sys\\_coeff\\_table, 11](#)
- [fcip\\_demand\\_sys\\_effect, 11](#)
- [fcip\\_demand\\_sys\\_estimate, 12](#)
- [fcip\\_demand\\_sys\\_fit, 13](#)
- [fcip\\_demand\\_sys\\_partial, 14](#)
- [fcip\\_demand\\_sys\\_prep, 15](#)
- [fcip\\_demand\\_sys\\_run, 15](#)
- [fcip\\_demand\\_sys\\_tests, 16](#)
- [fcip\\_demand\\_sys\\_vcov, 16](#)
- [fcip\\_recodes\\_commodity\\_groupings, 17](#)
- [fcip\\_recodes\\_insurance\\_plan, 18](#)
- [fcip\\_recodes\\_practice, 18](#)
- [fcip\\_recodes\\_type, 19](#)
- [fixed\\_effect\\_model\\_data\\_prep, 19](#)
- [format\\_fcip\\_demand\\_table, 20](#)
- [fsa\\_crop\\_linker, 21](#)
- [get\\_yu2018\\_instrument, 8, 21](#)
- [global\\_variables, 22](#)
- [nassAgPriceMonthlyIndex, 22](#)
- [nassSurveyMYAprice, 23](#)
- [nassSurveyPriceRecivedIndex, 23](#)
- [nassSurveyRentalRates, 24](#)
- [premium\\_subsidy\\_schedule, 24](#)
- [prevented\\_planting\\_adm, 25](#)