

Package: uscogdata (via r-universe)

October 11, 2026

Type Package

Title Curated Reader for the Civilytics US Census of Governments
Finance Corpus

Version 0.4.0

Description Curated R verbs over the Civilytics US Census of
Governments finance corpus. Provides unit-level financial
profiles, geographic rollups, and peer comparisons with
auditable provenance and built-in cross-vintage correctness.

License MIT + file LICENSE

URL <https://github.com/civilytics/uscogdata>,
<https://civilytics.r-universe.dev/uscogdata>

BugReports <https://github.com/civilytics/uscogdata/issues>

Encoding UTF-8

LazyData false

Depends R (>= 4.1)

Imports DBI (>= 1.1.0), duckdb (>= 1.0.0), dplyr (>= 1.1.0), tibble,
cli, jsonlite, httr2, digest

Suggests testthat (>= 3.0.0), withr, knitr, rmarkdown, pkgdown,
ggplot2

Config/testthat/edition 3

VignetteBuilder knitr

RoxygenNote 7.3.3

MinCorpusSchema 4

MaxCorpusSchema 7

Config/pak/sysreqs libssl-dev xz-utils

Repository <https://civilytics.r-universe.dev>

Date/Publication 2026-08-10 23:36:42 UTC

RemoteUrl <https://github.com/civilytics/uscogdata>

RemoteRef v0.4.0

RemoteSha d2caa6de970134890b97def3e882dd00bc68daa3

Contents

cog_balances	2
cog_basket_resolution	4
cog_basket_unresolved	5
cog_categories	6
cog_explain	6
cog_find_peers	7
cog_geographic_rollup	8
cog_gov_search	10
cog_manifest	13
cog_mirror	13
cog_peer_compare	14
cog_recipes	16
cog_revenue	16
cog_spending	19

Index	24
--------------	-----------

cog_balances	<i>Cash and security holdings for one or more governments</i>
--------------	---

Description

Returns Census cash-and-security holdings (`category_type = "balance"`): fund balances, retirement system holdings and insurance trust balances.

Usage

```
cog_balances(
  govid = NULL,
  years,
  category = NULL,
  per_capita = FALSE,
  adjust_to_year = NULL,
  basis = c("harmonized", "raw"),
  recipe = NULL,
  state = NULL,
  type = NULL,
  limit = NULL,
  offset = NULL
)
```

Arguments

<code>govid</code>	Canonical govid(s): a character vector, or a data frame with a ‘canonical_govid’ column (e.g. from <code>[cog_gov_search()]</code>). ‘NULL’ to name the cohort by ‘state’/‘type’ instead.
--------------------	--

years	Integer vector of fiscal years.
category	Optional character vector of categories to keep. One of "Fund Balances", "Insurance Trust Balances", "Retirement System Holdings". There is deliberately no 'subtype' argument: for holdings, 'category' is a strict coarsening of 'balance_subtype' (unlike the money verbs, where the two axes cross), so every combination would be either redundant or empty. 'category = "Fund Balances"' is exactly the 'general' family ('W01'/'W31'/'W61'). 'balance_subtype' is returned, so a finer split is one 'dplyr::filter()' away. The reserved pseudo-category "All Categories" (see [cog_spending()]) is not supported here and errors with class 'uscogdata_all_categories_unsupported': it sums a concept's subtype scope, and holdings are a stock with no concept vocabulary to sum across. Omit 'category' to get every category broken out instead.
per_capita	Divide holdings by population. Note this is a stock per resident (reserves per person), which is not comparable to [cog_spending()]'s per-capita figures – those are a flow per person.
adjust_to_year	Deflate to this year's dollars (CPI-U).
basis	Accepted for uniformity with the money verbs, but currently a no-op : 'harmonization_map' carries no balance-code rows, so harmonized and raw space are identical for holdings. Reported in 'provenance\$basis_note'.
recipe	Optional harmonization recipe id (see [cog_recipes()]). "cash_securities_z77_wide" and "cash_securities_z78_wide" bridge the wide era to the modern one.
state, type	Name the cohort by predicate instead of by id: 'state' is a 2-letter USPS abbreviation (or a FIPS code) and 'type' is one of "state", "county", "city", "township" (or the integer '0:3') – the same vocabulary, and the same internal coercion, as [cog_gov_search()]. Both default to 'NULL'. The cohort is then expressed as a subquery against 'canonical_fips_xwalk' inside each statement rather than round-tripped through R as a literal id list. For a fleet-scale cohort that is the difference between a 301,591-character 'IN' list re-parsed in 5–8 statements per call and a constant-size predicate: measured at 94 ms versus 449 ms for the same FY2022 aggregate over the 20,106-government 'type = "city"' cohort, within 7 Supplying 'govid' and 'state'/'type' INTERSECTS them – the governments in 'govid' that also match the predicate – rather than one silently taking precedence. Naming no cohort at all ('govid', 'state' and 'type' all 'NULL') aborts with class 'uscogdata_no_cohort'. When the cohort is named by predicate, 'provenance\$scope\$govids_found' and 'govids_missing' are empty – there is no id list to report against – and 'provenance\$scope\$cohort' carries 'state', 'type' and 'n_governments' instead. A 'govid'-named cohort reports exactly as before.
limit	Maximum number of result rows to return, pushed into the SQL rather than applied after materializing every row. 'NULL' (default) returns everything. Cannot be combined with 'recipe' – see 'offset' and 'total_rows'.
offset	Rows to skip before 'limit' starts counting (0-based). Ignored if 'limit' is 'NULL'; defaults to '0L' when 'limit' is set.

Value

Tibble with columns ‘year’, ‘canonical_govid’, ‘gov_name’, ‘balance_subtype’, ‘category’, ‘amt_nominal’, ‘codes_included’, ‘aggregate_fallback’, plus optional ‘amt_per_capita_nominal’ and ‘pop_source’ (when ‘per_capita = TRUE’), optional ‘amt_real’ (when ‘adjust_to_year’ is set), and optional ‘amt_per_capita_real’ (only when **both** ‘per_capita = TRUE’ and ‘adjust_to_year’ are set – there is no nominal per-capita column to deflate otherwise). Amounts are full US dollars.

Carries a ‘provenance’ attribute matching ‘inst/schemas/provenance-v1.json’, whose ‘balance_caveats’ block reports ‘not_gaap’, ‘not_gaap_note’, ‘coverage_window’ (measured year extents for every balance subtype in the mounted corpus, not only the observed ones) and ‘truncated’ (the observed subtypes whose coverage falls short of the requested years). ‘expenditure_concept’/‘revenue_concept’ are ‘NA’ – holdings are a stock, not a flow, so neither concept vocabulary applies.

When ‘limit’ is set, also carries a ‘total_rows’ attribute: the full unpaginated row count, computed by the same query (‘COUNT(*) OVER()’) rather than a second scan.

Holdings are not GAAP fund balance

Census holdings are **gross** – no liabilities are netted – so a reserve ratio built from them overstates what is actually available. They are not comparable to a GAAP fund balance from an ACFR.

cog_basket_resolution

Inspect basket-mode resolution sidecar

Description

Returns the resolution tibble attached to a basket-mode result of [cog_gov_search()]. One row per input entry; ‘status’ is one of “resolved”, “largest_pop”, “ambiguous”, “no_match”. By default the ‘candidates’ list-column is dropped for readable printing; pass ‘expand_candidates = TRUE’ to keep it.

Usage

```
cog_basket_resolution(x, expand_candidates = FALSE)
```

Arguments

x A tibble returned by basket-mode [cog_gov_search()].

expand_candidates Logical. If ‘TRUE’, keeps the ‘candidates’ list-column (full-schema match candidates per input row). Default ‘FALSE’.

Value

A tibble with the resolution audit trail.

Examples

```
## Not run:
basket <- cog_gov_search(
  name = c("Broward", "San Diego", "Notarealplace"),
  state = c("FL",      "CA",      "NY")
)
cog_basket_resolution(basket)
cog_basket_resolution(basket, expand_candidates = TRUE)

## End(Not run)
```

`cog_basket_unresolved`*Filter a basket resolution to unresolved rows*

Description

Convenience wrapper that returns just the rows where ‘status’ is ‘”ambiguous”‘ or ‘”no_match”‘ — the ones the user likely wants to refine before piping into a query verb. The ‘candidates’ list-column is preserved so the user can drill into ambiguous match sets.

Usage

```
cog_basket_unresolved(x)
```

Arguments

`x` A tibble returned by basket-mode `[cog_gov_search()]`.

Value

A tibble (subset of `[cog_basket_resolution()]`).

Examples

```
## Not run:
basket <- cog_gov_search(
  name = c("Broward", "San Diego", "Notarealplace"),
  state = c("FL",      "CA",      "NY")
)
cog_basket_unresolved(basket)

## End(Not run)
```

cog_categories	List available spending / revenue categories
----------------	--

Description

Returns the category taxonomy exposed by the corpus’s ‘summary_categories’ view, grouped to one row per ‘(category, subtype)’ pair. Use this to discover valid ‘category’ values for [cog_spending()] / [cog_revenue()] / [cog_balances()] / [cog_geographic_rollup()] and to audit which Census item codes feed each category.

Usage

```
cog_categories(type = NULL, pattern = NULL)
```

Arguments

- type** Either ‘NULL’ (default, every row: expenditure, revenue and balance), “spending”, “revenue”, or “balance”.
- pattern** Optional regex matched case-insensitively against the ‘category’ column (e.g. “Police” or “Tax”).

Details

‘subtype’ COALESCEs the crosswalk’s three subtype columns, so it carries ‘spend_subtype’ on expenditure rows, ‘revenue_subtype’ on revenue rows and ‘balance_subtype’ on balance rows. Note that [cog_balances()] itself takes no ‘subtype’ argument — for holdings, ‘category’ is a strict coarsening of ‘balance_subtype’ — but the value is surfaced here because it is the discovery surface downstream consumers build their vocabulary from.

Value

Tibble with columns ‘category’, ‘category_type’, ‘subtype’, ‘n_codes’, ‘item_codes’ (comma-separated, alphabetical). Sorted by ‘category_type’, ‘category’, ‘subtype’. Includes one row per flow for the reserved pseudo-category “All Categories”, which carries ‘NA’ for ‘subtype’, ‘n_codes’ and ‘item_codes’ because it is a query mode rather than a crosswalk entry — see [cog_spending()]’s ‘category’ argument.

cog_explain	Explain a verb result’s provenance
-------------	------------------------------------

Description

Prints the structured provenance attached to a tibble returned by any ‘cog_*’ verb, or returns it as a list for downstream use (MCP tools, dashboards, JSON export).

Usage

```
cog_explain(result, format = c("print", "list"))
```

Arguments

result A tibble returned by a ‘cog_*’ verb.

format “print” (default) for a human-readable cli summary; returns ‘result’ invisibly for chaining. “list” returns the raw provenance list (identical to ‘attr(result, "provenance")’).

Value

Either ‘result’ (invisibly) or the provenance list.

Two kinds of series break

Catalogued breaks reach you without being asked for, in two disjoint fields, because a caveat about one series and a caveat about the whole corpus are different claims:

* **‘series_break_refs’** — breaks matched against the item codes actually present in this result. A break in one code you queried. * **‘corpus_break_refs’** — breaks catalogued with ‘fin_code = "ALL"’, which are statements about the corpus rather than about any one code: dollar precision across the 1976/1977 boundary (‘SB085’), imputation exclusion from FY2002 (‘SB087’), the FY2012 dense-to-sparse representation change (‘SB194’), and the FY2017 government-identifier change (‘SB086’). These are selected on the break-year window alone.

‘SB194’ is the one most likely to matter: a query spanning FY2011 to FY2012 crosses the boundary where an absent cell stops meaning “Census published \$0” and starts meaning “not reported”.

Other provenance blocks

‘transformations\$units_conversion’ records the ‘\$1,000s’-to-dollars multiply that every amount column has already had applied. ‘transformations\$per_capita’ records the population denominator and its year range. ‘coverage’ and ‘coverage_mode’ appear on multi-government results (see [cog_geographic_rollup()]). ‘completion’ appears when ‘complete = TRUE’. ‘balance_caveats’ appears on [cog_balances()] results.

cog_find_peers

Find peer governments by similarity criteria

Description

Selects peer governments by combinations of government type, state, and population range at a chosen ‘year’. Peers are ordered by ‘|log(pop_ratio)|’ ascending (closest to the target’s population first).

Usage

```
cog_find_peers(
  target_govid,
  year = NULL,
  same_type = TRUE,
  same_state = FALSE,
  pop_range = c(0.7, 1.3),
  is_ratio = TRUE,
  max_peers = 10L,
  coverage = c("all", "census", "consistent")
)
```

Arguments

target_govid	Character scalar — ‘canonical_govid’ of the target.
year	Integer scalar. Cohort vintage. When ‘NULL’ (default), uses the most recent year for which the target has an observed population in ‘gov_population_yearly’.
same_type	If ‘TRUE’ (default) restrict peers to the target’s ‘govs_type’.
same_state	If ‘TRUE’ restrict peers to the target’s ‘fips_state’. Default ‘FALSE’.
pop_range	Length-2 numeric vector giving lower/upper bounds.
is_ratio	If ‘TRUE’ (default) ‘pop_range’ is multiplied by the target’s population at ‘year’ to produce absolute bounds. If ‘FALSE’, ‘pop_range’ is interpreted as absolute population counts.
max_peers	Integer cap on the number of peers returned.
coverage	Survey-cycle handling; see [cog_peer_compare()]. Here it governs the cohort VINTAGE when ‘year’ is ‘NULL’: “census” snaps to the most recent census year with an observed population, so a cohort is not built from a sample year in which most of the candidate universe is absent. “consistent” needs a year range, which cohort selection does not have, so it selects like “all” and is carried on the result as ‘attr(x, “coverage”)’ for [cog_peer_compare()].

Value

Tibble with columns ‘canonical_govid’, ‘gov_name’, ‘fips_state’, ‘population’, ‘pop_ratio’, ‘rank’. The cohort year is attached as ‘attr(x, “cohort_year”)’.

cog_geographic_rollup

Aggregate spending across state/county/city layers for a place

Description

Wraps [cog_spending()], tags each row with its layer, and attaches a human-readable ‘scope_note’ documenting geographic-scope caveats (e.g. “county totals include areas outside the listed city”). Useful for “place portraits” that compare a city to the surrounding county and containing state on one set of axes.

Usage

```
cog_geographic_rollup(
  govids,
  category,
  years,
  per_capita = FALSE,
  adjust_to_year = NULL,
  expenditure_concept = c("primary", "direct", "total"),
  coverage = c("all", "census", "consistent")
)
```

Arguments

- | | |
|----------------------------|---|
| govids | Named list with any non-empty subset of elements named ‘state’, ‘county’, ‘city’. Each element is a character vector of ‘canonical_govid’ values. At least one layer required. |
| category | Single category name or character vector (passed through to <code>[cog_spending()]</code>), or the reserved “All Categories” for one summed row per ‘(year, canonical_govid, subtype)’ covering every category in the concept’s scope. “All Categories” is the efficient way to build a geographic total: without it a caller must issue one rollup per category and sum the results themselves. |
| years | Integer vector of years. |
| per_capita | If ‘TRUE’, per-capita uses each gov’s own per-year population from ‘gov_population_yearly’. Govs with missing population are excluded from the result. |
| adjust_to_year | Integer base year for CPI-U conversion, or ‘NULL’. |
| expenditure_concept | <p>“primary” (default), “direct”, or “total” – see <code>[cog_spending()]</code> for the three concepts. “total” is refused here because combining Total across multiple layers of government double-counts intergovernmental transfers (a state’s payment to a school district is the same dollar the district reports as its own Direct spending); “primary” and “direct” combine safely.</p> |
| coverage | <p>How to handle the Census of Governments survey cycle, which is a complete census only in years ending in 2 and 7 – every other year is a sample, and the sample varies enormously (on the bundled fixture, Wisconsin’s 608-city universe reports 597 governments in FY2012 and 112 in FY2019).</p> <p>* “all” (default) – every unit that reported that year. Unchanged behaviour, so existing code keeps working. * “census” – census years only. Aborts if the requested range holds none, rather than silently returning nothing. * “consistent” – only units reporting in every requested year, giving a balanced panel.</p> <p>Regardless of mode, ‘provenance\$coverage’ always carries per-year ‘n_units_reporting’, ‘n_units_expected’ and ‘is_census_year’, and ‘provenance\$coverage_mode’ records the mode. ‘is_census_year’ is a statement about the survey</p> |

calendar**, never a claim of completeness: FY1967 is a census year in which only 97 of Wisconsin's 608 cities report. 'n_units_reporting' is the number that tells the truth.

Details

When 'per_capita = TRUE', rows whose government has no observed population in that year ('pop_source == "unavailable"') are dropped from the result. The dropped govids are recorded in 'provenance\$rollup\$excluded_govids'. This excludes special districts (gov type 4) and school districts (gov type 5) from per-capita rollups by design — see 'vignette("population-denominators")'.

Value

Tibble with columns 'year', 'layer', 'canonical_govid', 'gov_name', 'spend_subtype', 'category', 'amt_nominal', optional 'amt_real' / 'amt_per_capita_nominal' / 'amt_per_capita_real', optional 'pop_source', 'codes_included', 'aggregate_fallback', 'scope_note', 'notes'. Carries a 'provenance' attribute with 'verb = "cog_geographic_rollup"', 'layers', and 'rollup\$included_govids' / 'rollup\$excluded_govids'.

Reading 'coverage'

'provenance\$coverage' reports 'n_units_reporting' against 'n_units_expected' per year. **'n_units_reporting' is category-conditional: it counts governments with rows for the category you asked for, not governments collected that year.** A government that was surveyed and genuinely spends nothing in that category is indistinguishable here from one that was never surveyed.

The ratio is therefore **not a response rate** and must not be used as one. In FY2022 — a complete census year — Georgia reports 393 of 567 cities for 'category = "Police"'; the 174-city gap is overwhelmingly cities that contract policing to the county sheriff, not non-response.

The comparison that *is* valid is the same category across a census year (ending in 2 or 7) and a sample year, where the real-zero component is roughly constant and the difference reflects the survey cycle. 'is_census_year' marks which is which.

cog_gov_search

Search for governments by name, state, and/or type

Description

Resolves human-readable place names into rows of 'canonical_fips_xwalk', the cross-vintage canonical-government registry. Operates in two modes:

Usage

```
cog_gov_search(
  name = NULL,
  state = NULL,
  type = NULL,
  limit = NULL,
  offset = NULL
)
```

Arguments

name	Character vector of place name(s). Length 1 = utility mode; length >1 = basket mode.
state	2-letter USPS abbreviation (e.g. "FL"), FIPS integer (e.g. '12'), or 'NULL'. In basket mode, length 1 recycles across all entries; otherwise must match 'length(name)'.
type	Government type: integer in '0:3' or one of "state", "county", "city", "township", or 'NA'/'NULL'. Per-row optional in basket mode (recycles from length 1). Excluded types '4'/'5' (or "special_district" / "school_district") trigger an explanatory message and an empty result.
limit	Maximum number of rows to return, applied in SQL. 'NULL' (default) returns every match – which, with no other filter, is the entire crosswalk. Utility mode only: pagination has no meaning in basket mode, where the result is one resolved row per requested name in input order, and is refused there with class 'uscogdata_basket_pagination_conflict'.
offset	Rows to skip before 'limit' starts counting (0-based). Ignored if 'limit' is 'NULL'; defaults to '0L' when 'limit' is set.

Details

Utility mode (single 'name', the original behavior): returns all rows whose 'gov_name' contains 'name' as a **literal, case-insensitive substring**, sorted by 'population_acs' descending. Useful for exploratory lookups. Regex metacharacters in 'name' are escaped, so a government is findable by its own complete name even when that name contains parentheses or a period. **Basket mode** ('length(name) > 1'): resolves each input row to a single canonical govid and returns a tibble in input order, suitable for piping straight into [cog_spending()] / [cog_revenue()] / [cog_geographic_rollup()]. Carries an audit sidecar accessible via [cog_basket_resolution()] / [cog_basket_unresolved()].

Basket-mode resolution algorithm (per input row): 1. Filter 'canonical_fips_xwalk' by 'state' and (if non-NA) 'type'. 2. **Exact pass:** case-insensitive equality against 'gov_name'. Single hit -> resolved. Multiple -> step 4. 3. **Substring fallback:** case-insensitive literal substring against 'gov_name' (metacharacters escaped). Single hit -> resolved ('match_method = "substring"'). Zero hits -> 'status = "no_match"'. Multiple hits -> step 4. 4. **Disambiguation:** if matches share one 'govs_type', pick the largest-population row ('status = "largest_pop"'). If they span >=2 types, no row is added ('status = "ambiguous"'); the user should re-run with 'type' specified.

Resolved rows form the returned tibble in input order. Unresolved inputs ('ambiguous' / 'no_match') appear only in the sidecar.

Value

A tibble of 'canonical_fips_xwalk' rows. In utility mode, all matches sorted by 'population_acs' desc, ties broken by 'canonical_govid'. In basket mode, resolved rows in input order, with 'attr(, "resolution")' set to the sidecar tibble.

When 'limit' is set, carries a 'total_rows' attribute: the full unpaginated match count, computed by the same query ('COUNT(*) OVER()') rather than a second scan.

See Also

[cog_basket_resolution()], [cog_basket_unresolved()], [cog_spending()], [cog_revenue()].

Examples

```
## Not run:
# Utility mode - exploratory substring lookup
cog_gov_search("broward", state = "FL")

# Basket mode - resolve a known cohort
basket <- cog_gov_search(
  name = c("BROWARD COUNTY", "SAN DIEGO CITY", "AUSTIN CITY"),
  state = c("FL", "CA", "TX")
)
basket

# Inspect resolution audit
cog_basket_resolution(basket)

# Pipe into a spending query
library(dplyr)
basket |> cog_spending(years = 2019:2020, category = "Police")

# Iteratively refine ambiguous matches
partial <- cog_gov_search(
  name = c("Broward", "San Diego"), # San Diego is ambiguous
  state = c("FL", "CA")
)
cog_basket_unresolved(partial)
refined <- cog_gov_search(
  name = c("Broward", "San Diego"),
  state = c("FL", "CA"),
  type = c(NA, "city") # disambiguate
)

## End(Not run)
```

<code>cog_manifest</code>	<i>Return the parsed corpus manifest for the active session.</i>
---------------------------	--

Description

Opens a session (connecting to the configured corpus) if none is active, then returns the manifest exactly as parsed from ‘manifest.json’. Useful for consumers that need the published year range (‘years’ block, schema v5+) or the partition list without issuing a data query.

Usage

```
cog_manifest()
```

Value

Named list: ‘schema_version’, ‘built_at’, ‘pipeline_commit’, ‘data_vintage’, ‘scope’, ‘years’ (schema v5+), ‘schema’, ‘files’.

<code>cog_mirror</code>	<i>Mirror the published corpus to a local directory</i>
-------------------------	---

Description

Downloads (or copies, for local-path fixture URLs) every file listed in the session’s ‘manifest.json’ into ‘dest’, preserving the relative path structure. Files already present with a matching SHA-256 hash are skipped unless ‘overwrite = TRUE’. The resulting directory can be pointed at via ‘options(uscogdata.url = paste0(dest, "/"))’ for offline analysis.

Usage

```
cog_mirror(
  dest,
  include = c("long", "metadata", "docs"),
  overwrite = FALSE,
  progress = interactive()
)
```

Arguments

<code>dest</code>	Destination directory. Created if missing.
<code>include</code>	Subset of ‘c("long", "metadata", "docs")’ controlling which manifest sections to mirror.
<code>overwrite</code>	If ‘TRUE’, re-copy even when the local SHA matches.
<code>progress</code>	If ‘TRUE’, show a cli progress bar.

Value

Invisibly, a tibble of `(path, sha256, size_bytes, status)` rows where `'status'` is `"downloaded"` or `"cached"`.

<code>cog_peer_compare</code>	<i>Compare a target government against a peer set</i>
-------------------------------	---

Description

Pulls spending for the target plus a peer set (either a `[cog_find_peers()]` result or a character vector of `'canonical_govid'`) and appends peer-distribution summary rows (`'summary_p25'`, `'summary_p50'`, `'summary_p75'`) so the result can be faceted by `'role'` in a single ggplot call. Those summary rows are quantiles ***within each category***, not quantiles of each peer's total — see the `'@return'` section before summing them.

Usage

```
cog_peer_compare(
  target_govid,
  peers,
  category,
  years,
  per_capita = TRUE,
  adjust_to_year = NULL,
  expenditure_concept = c("primary", "direct", "total"),
  coverage = c("all", "census", "consistent")
)
```

Arguments

<code>target_govid</code>	Character scalar.
<code>peers</code>	A tibble from <code>[cog_find_peers()]</code> or a character vector of <code>'canonical_govid'</code> s.
<code>category</code>	Character scalar or vector.
<code>years</code>	Integer vector.
<code>per_capita</code>	Default <code>'TRUE'</code> — peer compare usually normalizes by population.
<code>adjust_to_year</code>	Integer base year for CPI-U conversion or <code>'NULL'</code> .
<code>expenditure_concept</code>	<code>"primary"</code> (default), <code>"direct"</code> , or <code>"total"</code> — see <code>[cog_spending()]</code> for the three concepts. <code>"total"</code> is refused here because combining Total across peer sets counts intergovernmental transfers twice; <code>"primary"</code> and <code>"direct"</code> combine safely.

coverage

How to handle the Census of Governments survey cycle, which is a **complete census only in years ending in 2 and 7** – every other year is a sample, and the sample varies enormously (on the bundled fixture, Wisconsin’s 608-city universe reports 597 governments in FY2012 and 112 in FY2019).

“all” (default) – every unit that reported that year. Unchanged behaviour, so existing code keeps working. **“census”** – census years only. Aborts if the requested range holds none, rather than silently returning nothing. **“consistent”** – only units reporting in **every** requested year, giving a balanced panel.

Regardless of mode, `provenance$coverage` always carries per-year `n_units_reporting`, `n_units_expected` and `is_census_year`, and `provenance$coverage_mode` records the mode. `is_census_year` is a statement about the **survey calendar**, never a claim of completeness: FY1967 is a census year in which only 97 of Wisconsin’s 608 cities report. `n_units_reporting` is the number that tells the truth.

The comparison target is exempt from **“consistent”** balancing – it is the subject of the comparison, not a member of the cohort – and the `summary_*` quantiles are computed **AFTER** the filter, so they describe the cohort actually returned. `n_units_reporting` counts peers only, against the cohort size: **“3 of your 15 peers reported in FY2019”**.

Value

Tibble matching `[cog_spending()]`’s columns, plus a `role` column taking values **“target”**, **“peer”**, **“summary_p25”**, **“summary_p50”**, or **“summary_p75”**, `target_rank` (target’s rank among target+peers at `max(years)`, NA for other rows), and `cohort_year` (the year used to build the peer cohort, read from `attr(peers, “cohort_year”)`; NA when `peers` was a bare character vector). Provenance reports `verb = “cog_peer_compare”`, `peer_count`, `cohort_year`, and `cohort_govids`.

The ‘summary_’ rows are per-category quantiles: they are not additive. Each one is computed **within each (year, spend_subtype, category) cell** across the peer set, so a `summary_p50` row is **the median peer’s value in that one category**, not **the value of the median peer’s total**. The median peer for Police and the median peer for Fire are usually different governments, so summing `summary_*` rows across categories does not give any peer’s total and misstates the band it appears to describe — measured at **−32.7 one cohort**, with a sign flip at FY2012.

Facet by `role` **and** `category` (the documented use, and what the rows are built for). For a genuine **“median peer’s total spending”** line, sum each peer’s own categories first and take the quantile of those per-government totals:

```
“r library(dplyr) cmp |> filter(role group_by(year, role, canonical_govid) |> summarise(total
= sum(amt_per_capita_real, na.rm = TRUE), .groups = “drop”) |> filter(role == “peer”)
|> group_by(year) |> summarise(p50 = quantile(total, 0.5, na.rm = TRUE)) “
```

Reading ‘coverage’

`provenance$coverage` reports `n_units_reporting` against `n_units_expected` per year. **`n_units_reporting` is category-conditional: it counts cohort members with rows for the**

category you asked for, not cohort members collected that year.** A government that was surveyed and genuinely spends nothing in that category is indistinguishable here from one that was never surveyed.

The ratio is therefore **not** a response rate and must not be used as one. In FY2022 — a complete census year — Georgia reports 393 of 567 cities for ‘category = "Police"’; the 174-city gap is overwhelmingly cities that contract policing to the county sheriff, not non-response.

The comparison that **is** valid is the same category across a census year (ending in 2 or 7) and a sample year, where the real-zero component is roughly constant and the difference reflects the survey cycle. ‘is_census_year’ marks which is which.

cog_recipes	List available harmonization recipes
-------------	--------------------------------------

Description

Recipes are multi-code cross-vintage series (see [cog_spending()])’s ‘recipe’ argument) catalogued in the corpus’s ‘harmonization_recipes’ table. Use this to discover valid ‘recipe’ ids.

Usage

cog_recipes(pattern = NULL)

Arguments

pattern Optional regex matched case-insensitively against ‘recipe_id’ or ‘label’.

Value

Tibble with columns ‘recipe_id’, ‘label’, ‘n_components’, ‘year_min’, ‘year_max’ (the min/max component year coverage), sorted by ‘recipe_id’.

cog_revenue	Summarized revenue by category
-------------	--------------------------------

Description

Mirror of [cog_spending()] for revenue categories. One row per ‘(year, canonical_govid, revenue_subtype, category)’. Amounts are returned in **full U.S. dollars** (raw Census values are in \$1,000s; this verb multiplies by 1000 and records the conversion in ‘provenance’).

Usage

```
cog_revenue(
  govid = NULL,
  years,
  category = NULL,
  per_capita = FALSE,
  adjust_to_year = NULL,
  basis = c("harmonized", "raw"),
  recipe = NULL,
  revenue_concept = c("general", "total"),
  complete = FALSE,
  limit = NULL,
  offset = NULL,
  state = NULL,
  type = NULL
)
```

Arguments

govid	Character vector of ‘canonical_govid’ values, or ‘NULL’ to name the cohort by ‘state’/‘type’ instead. One of ‘govid’, ‘state’, or ‘type’ is required.
years	Integer vector of years.
category	Character vector of category names (from ‘summary_categories.category’), or ‘NULL’ for all categories broken out one row each. The reserved value “All Categories” instead returns a single summed row per ‘(year, canonical_govid, subtype)’, covering every category inside the requested concept’s subtype scope. It cannot be combined with other category names, and it is not the same thing as ‘revenue_concept = "total"’: the concept chooses which subtypes are in scope, “All Categories” chooses whether rows inside that scope are broken out or summed. Because the result keeps one row per ‘revenue_subtype’, filtering the returned frame to ‘revenue_subtype == "own_source"’ gives an own-source revenue total.
per_capita	If ‘TRUE’, adds ‘amt_per_capita_nominal’ (and ‘amt_per_capita_real’ when ‘adjust_to_year’ is set) using the per-year Census F-33 population from ‘gov_population_yearly’. Result also gains a ‘pop_source’ column with values “census_f33” or “unavailable” (the latter for gov types 4/5 and any row whose population is missing in that year).
adjust_to_year	Integer base year for CPI-U real-dollar conversion, or ‘NULL’ for nominal only.
basis	“harmonized” (default) sums item codes through the cross-vintage harmonization mapping (folding series-break-affected codes onto a comparable target and excluding aggregate / discontinued rows – see the ‘harmonization’ block in ‘cog_explain()’); “raw” reproduces the pre-Phase-R2 behavior (published item codes, no folding). On a corpus with ‘schema_version < 5’ (no harmonization tables), ‘basis’ silently resolves to “raw” when

	left at its default and the resolution is recorded in the provenance; explicitly passing 'basis = "harmonized"' on such a corpus aborts. Ignored when 'recipe' is set (see below).
recipe	Optional harmonization recipe id (see [cog_recipes()]) for multi-code cross-vintage series that a 1:1 harmonized_code mapping can't express (e.g. a wide-era aggregate that only splits into leaf codes in the modern era). Mutually exclusive with 'category'. The result's subtype column reads "recipe" and 'category' reads the recipe's label. Requires 'schema_version >= 5'. A recipe query bypasses 'basis' entirely (it joins 'long' directly rather than going through the '*_annotated'/'*_annotated_harmonized' views), so the 'basis' argument is ignored and the result's provenance reports 'basis = "recipe"' with an inert 'harmonization' block ('applied = FALSE', pointing at the 'recipe' block instead) rather than a possibly-misleading "harmonized"/"raw" value.
revenue_concept	Which of Census's two published revenue concepts to return. Concepts are defined as sets of the crosswalk's 'revenue_subtype' values – never as item-code first letters, which cannot classify correctly (prefix 'Y' spans revenue, expenditure and balance codes, and prefix 'X' does the same): * "general" (default) – Census General Revenue: 'own_source' + 'federal' + 'state' + 'local_aid'. The manual defines this concept by subtraction (section 4.3: "General revenue comprises all revenue except that classified as liquor store, utility, or insurance trust revenue"), so utility ('A91'-'A94'), liquor store ('A90') and insurance trust revenue are all excluded. * "total" – Census Total Revenue: every revenue subtype, i.e. 'general' plus utility, liquor store, and insurance trust revenue ('Y01'/'Y02'/'Y04'/'Y11'/'Y12'/'Y51'/'Y52' and the employee-retirement 'X01'/'X02'/'X05'/'X08'). The two are related by Census's own identity, 'Total Revenue = General + Utility + Liquor Store + Insurance Trust'. Note that the employee-retirement ('X') codes stop at FY2016, when those systems moved out of the annual finance file into the separate Annual Survey of Public Pensions, so a "total" series steps down at the FY2016/FY2017 seam for reasons that are about collection scope rather than revenue (series breaks 'SB197'-'SB202').
complete	If 'TRUE', fill the requested grid so that a cell the corpus does not carry still appears, labelled with **why** it is missing, and add a 'value_source' column to every row: * "reported" — the corpus carries this cell. * "census_zero" — dense-source year ('<= FY2011'), cell absent: Census published '\$0'. 'amt_nominal' is '0'. * "not_reported" — sparse-source year ('>= FY2012'), cell absent: the government did not report, and the value is unknown. 'amt_nominal' is 'NA', **not** '0' — writing a zero there would invent data. The grid comes from the corpus's 'code_set' table, scoped to each government's own type, so a county is never filled with cells only a state can report. Reported rows are passed through untouched. Defaults to 'FALSE' (the historical behaviour: absent cells simply do not appear). Needs a corpus published from 2026-07-29 onward, which

	is when ‘representation’/‘code_set’ began shipping; aborts with class ‘uscogdata_representation_unavailable’ otherwise. Not available with ‘recipe’ or with ‘expenditure_concept = ”total”’ (class ‘uscogdata_complete_unsupported’) — neither draws its cells from ‘code_set’.
limit	Maximum number of result rows to return, pushed into the SQL query itself (‘LIMIT’/‘OFFSET’) rather than applied after the full result is materialized. ‘NULL’ (the default) returns every matching row, exactly as before this parameter existed. Mutually exclusive with ‘recipe’ and with ‘complete = TRUE’ – see ‘offset’ and ‘total_rows’.
offset	Rows to skip before ‘limit’ starts counting (0-based). Ignored if ‘limit’ is ‘NULL’; defaults to ‘0L’ when ‘limit’ is set.
state, type	Name the cohort by predicate instead of by id: ‘state’ is a 2-letter USPS abbreviation (or a FIPS code) and ‘type’ is one of “state”, “county”, “city”, “township” (or the integer ‘0:3’) – the same vocabulary, and the same internal coercion, as [cog_gov_search()]. Both default to ‘NULL’. The cohort is then expressed as a subquery against ‘canonical_fips_xwalk’ inside each statement rather than round-tripped through R as a literal id list. For a fleet-scale cohort that is the difference between a 301,591-character ‘IN’ list re-parsed in 5–8 statements per call and a constant-size predicate: measured at **94 ms versus 449 ms for the same FY2022 aggregate over the 20,106-government ‘type = ”city”’ cohort, within 7 Supplying ‘govid’ **and** ‘state’/‘type’ INTERSECTS them – the governments in ‘govid’ that also match the predicate – rather than one silently taking precedence. Naming no cohort at all (‘govid’, ‘state’ and ‘type’ all ‘NULL’) aborts with class ‘uscogdata_no_cohort’. When the cohort is named by predicate, ‘provenance\$scope\$govids_found’ and ‘govids_missing’ are empty – there is no id list to report against – and ‘provenance\$scope\$cohort’ carries ‘state’, ‘type’ and ‘n_governments’ instead. A ‘govid’-named cohort reports exactly as before.

Value

Tibble with columns ‘year’, ‘canonical_govid’, ‘gov_name’, ‘revenue_subtype’, ‘category’, ‘amt_nominal’, optional ‘amt_real’, optional ‘amt_per_capita_nominal’, optional ‘amt_per_capita_real’, optional ‘pop_source’, ‘codes_included’, ‘aggregate_fallback’, ‘notes’, and ‘value_source’ when ‘complete = TRUE’.

cog_spending

Summarized spending by category

Description

One row per ‘(year, canonical_govid, spend_subtype, category)’. Amounts are returned in ****full U.S. dollars**** (the raw corpus stores them in \$1,000s; this verb multiplies by 1000 so downstream code can freely rescale to millions/billions). The conversion is recorded in the provenance attribute under ‘transformations\$units_conversion’.

Usage

```
cog_spending(
  govid = NULL,
  years,
  category = NULL,
  per_capita = FALSE,
  adjust_to_year = NULL,
  basis = c("harmonized", "raw"),
  recipe = NULL,
  expenditure_concept = c("primary", "direct", "total"),
  complete = FALSE,
  limit = NULL,
  offset = NULL,
  state = NULL,
  type = NULL
)
```

Arguments

govid	Character vector of ‘canonical_govid’ values, or ‘NULL’ to name the cohort by ‘state’/‘type’ instead. One of ‘govid’, ‘state’, or ‘type’ is required.
years	Integer vector of years.
category	Character vector of category names (from ‘summary_categories.category’), or ‘NULL’ for all categories broken out one row each. The reserved value “All Categories” instead returns a single summed row per ‘(year, canonical_govid, subtype)’, covering every category inside the requested concept’s subtype scope. It cannot be combined with other category names, and it is not the same thing as ‘expenditure_concept = "total"’: the concept chooses which subtypes are in scope, “All Categories” chooses whether rows inside that scope are broken out or summed. Because the result keeps one row per ‘spend_subtype’, filtering the returned frame to ‘spend_subtype == "operations"’ gives an operating-expenditure total.
per_capita	If ‘TRUE’, adds ‘amt_per_capita_nominal’ (and ‘amt_per_capita_real’ when ‘adjust_to_year’ is set) using the per-year Census F-33 population from ‘gov_population_yearly’. Result also gains a ‘pop_source’ column with values “census_f33” or “unavailable” (the latter for gov types 4/5 and any row whose population is missing in that year).
adjust_to_year	Integer base year for CPI-U real-dollar conversion, or ‘NULL’ for nominal only.
basis	“harmonized” (default) sums item codes through the cross-vintage harmonization mapping (folding series-break-affected codes onto a comparable target and excluding aggregate / discontinued rows – see the ‘harmonization’ block in ‘cog_explain()’); “raw” reproduces the pre-Phase-R2 behavior (published item codes, no folding). On a corpus with ‘schema_version < 5’ (no harmonization tables), ‘basis’ silently resolves to “raw” when

left at its default and the resolution is recorded in the provenance; explicitly passing 'basis = "harmonized"' on such a corpus aborts. Ignored when 'recipe' is set (see below).

recipe Optional harmonization recipe id (see [cog_recipes()]) for multi-code cross-vintage series that a 1:1 harmonized_code mapping can't express (e.g. a wide-era aggregate that only splits into leaf codes in the modern era). Mutually exclusive with 'category'. The result's subtype column reads "recipe" and 'category' reads the recipe's label. Requires 'schema_version >= 5'. A recipe query bypasses 'basis' entirely (it joins 'long' directly rather than going through the '*_annotated'/'*_annotated_harmonized' views), so the 'basis' argument is ignored and the result's provenance reports 'basis = "recipe"' with an inert 'harmonization' block ('applied = FALSE', pointing at the 'recipe' block instead) rather than a possibly-misleading "harmonized"/"raw" value.

expenditure_concept

Which spending concept to return. Concepts are defined as sets of the crosswalk's 'spend_subtype' values – never as item-code first letters, which cannot classify correctly (prefix 'Y' alone spans revenue, expenditure, and balance codes):

* "primary" (default) – the government's own service provision: 'operations' + 'capital' + 'assistance' subtypes. * "direct" – Census's published Direct Expenditure: 'primary' plus 'interest' (interest on debt) and 'insurance_benefits' (insurance trust benefit payments, e.g. pensions – Census manual section 5.2.2.1 includes payments to retirees in Direct). * "total" – 'direct' plus the intergovernmental leg: payments to local governments ('M' codes), to the state government ('L' codes, excluding the 'L-' family-total rollup), and state payments to school systems ('Q11'/'Q12'/'Q18'), so results gain rows with 'spend_subtype == "intergovernmental"'. Requires the active corpus's 'summary_categories' to carry M/L rows (added by cog_pipeline PR #59); aborts with class 'uscogdata_ig_categories_unsupported' on an older corpus rather than silently under-reporting. Mutually exclusive with 'recipe' (a recipe already defines its own component codes).

****Do not sum "total" results across levels of government**** (e.g. state + county + city): a state's 'M12' payment to a school district is the same dollar the district reports as its own direct 'E12', so summing both double-counts it. This matters in particular with [cog_geographic_rollup()], which sums across exactly that kind of multi-layer government set.

In the legacy wide era (<= FY2011), some functions are published ONLY as an aggregate-flagged family total (e.g. Corrections' 'E04'/'E05' split), which the Direct leg excludes by construction but the IG leg deliberately keeps (see 'inst/sql/24-ig_long.sql'). For a "total" query, any (year, category) where this leaves intergovernmental rows with NO Direct counterpart is flagged: the affected rows' 'notes' name the harmonization recipe that recovers the missing Direct component (when one exists), and 'provenance\$expenditure_concept_direct_suppressed' is 'TRUE' – the figure in those rows is the intergovernmental leg alone, not Direct + IG. When 'category = "All Categories"' is combined with 'expen-

	diture_concept = "total", this detection cannot run (it keys on per-category rows, which all-categories mode collapses to one literal value), so 'expenditure_concept_direct_suppressed' is 'NA' rather than a possibly-false 'FALSE'; query an explicit 'category' to get a real answer.
complete	If 'TRUE', fill the requested grid so that a cell the corpus does not carry still appears, labelled with **why** it is missing, and add a 'value_source' column to every row: * "reported" — the corpus carries this cell. * "census_zero" — dense-source year ('<= FY2011'), cell absent: Census published '\$0'. 'amt_nominal' is '0'. * "not_reported" — sparse-source year ('>= FY2012'), cell absent: the government did not report, and the value is unknown. 'amt_nominal' is 'NA', **not** '0' — writing a zero there would invent data. The grid comes from the corpus's 'code_set' table, scoped to each government's own type, so a county is never filled with cells only a state can report. Reported rows are passed through untouched. Defaults to 'FALSE' (the historical behaviour: absent cells simply do not appear). Needs a corpus published from 2026-07-29 onward, which is when 'representation'/'code_set' began shipping; aborts with class 'uscogdata_representation_unavailable' otherwise. Not available with 'recipe' or with 'expenditure_concept = "total"' (class 'uscogdata_complete_unsupported') — neither draws its cells from 'code_set'.
limit	Maximum number of result rows to return, pushed into the SQL query itself ('LIMIT'/'OFFSET') rather than applied after the full result is materialized. 'NULL' (the default) returns every matching row, exactly as before this parameter existed. Mutually exclusive with 'recipe' and with 'complete = TRUE' – see 'offset' and 'total_rows'.
offset	Rows to skip before 'limit' starts counting (0-based). Ignored if 'limit' is 'NULL'; defaults to '0L' when 'limit' is set.
state, type	Name the cohort by predicate instead of by id: 'state' is a 2-letter USPS abbreviation (or a FIPS code) and 'type' is one of "state", "county", "city", "township" (or the integer '0:3') – the same vocabulary, and the same internal coercion, as [cog_gov_search()]. Both default to 'NULL'. The cohort is then expressed as a subquery against 'canonical_fips_xwalk' inside each statement rather than round-tripped through R as a literal id list. For a fleet-scale cohort that is the difference between a 301,591-character 'IN' list re-parsed in 5–8 statements per call and a constant-size predicate: measured at **94 ms versus 449 ms** for the same FY2022 aggregate over the 20,106-government 'type = "city"' cohort, within 7 Supplying 'govid' **and** 'state'/'type' INTERSECTS them – the governments in 'govid' that also match the predicate – rather than one silently taking precedence. Naming no cohort at all ('govid', 'state' and 'type' all 'NULL') aborts with class 'uscogdata_no_cohort'. When the cohort is named by predicate, 'provenance\$scope\$govids_found' and 'govids_missing' are empty – there is no id list to report against – and 'provenance\$scope\$cohort' carries 'state', 'type' and 'n_governments' instead. A 'govid'-named cohort reports exactly as before.

Value

Tibble with columns ‘year’, ‘canonical_govid’, ‘gov_name’, ‘spend_subtype’, ‘category’, ‘amt_nominal’, optional ‘amt_real’, optional ‘amt_per_capita_nominal’, optional ‘amt_per_capita_real’, optional ‘pop_source’, ‘codes_included’, ‘aggregate_fallback’, ‘notes’, and ‘value_source’ when ‘complete = TRUE’. Carries a ‘provenance’ attribute matching ‘inst/schemas/provenance-v1.json’, whose ‘completion’ block reports ‘applied’, ‘rows_filled’, and the per-year ‘absence_means’ rule that was applied. When ‘limit’ is set, also carries a ‘total_rows’ attribute: the full unpaginated row count, computed by the same query (‘COUNT(*) OVER()’) rather than a second round trip – so a caller walking pages never has to ask “how many are there” separately.

Index

cog_balances, [2](#)
cog_basket_resolution, [4](#)
cog_basket_unresolved, [5](#)
cog_categories, [6](#)
cog_explain, [6](#)
cog_find_peers, [7](#)
cog_geographic_rollup, [8](#)
cog_gov_search, [10](#)
cog_manifest, [13](#)
cog_mirror, [13](#)
cog_peer_compare, [14](#)
cog_recipes, [16](#)
cog_revenue, [16](#)
cog_spending, [19](#)