

Package: shellgame (via r-universe)

May 28, 2026

Title The Shell Game - Audit Geographic Data Transformations

Version 0.1.0

Description Reveals how data quality silently degrades during geographic transformations while variable labels remain unchanged. Demonstrates that transformation error is agnostic to both the variable (population, income, etc.) and the tool (R, Python, etc.). Provides a reproducible audit framework for quantifying the shift from observed to imputed data at each transformation hop.

License MIT + file LICENSE

URL <https://github.com/phinnphace/shellgame>

BugReports <https://github.com/phinnphace/shellgame/issues>

Depends R (>= 4.0.0)

Imports dplyr (>= 1.0.0), ggplot2, janitor, magrittr, rlang, stringr, tidycensus, utils

Suggests geoDeltaAudit, knitr, readr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

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

Date/Publication 2026-05-17 19:15:57 UTC

RemoteUrl <https://github.com/phinnphace/shellgame>

RemoteRef HEAD

RemoteSha 3e3076b48c9628c28882dde5bb2c078cbde66f13

Contents

audit_transformation	2
check_census_key	3
create_audit_report	3
extract_perturbed_population	4
get_zcta_baseline	5
pad_geoid	5
plot_baseline_zctas	6
plot_geometric_vs_relationship	6
plot_transformation_perturbation	7
prep_hud_crosswalk	8
prep_zip_zcta	8
print.shellgame_audit	9
run_full_transformation	9
summary.shellgame_audit	10
transform_zcta_to_zip	11
transform_zip_to_county	11

Index	13
--------------	-----------

audit_transformation *Audit geographic transformation*

Description

Main function to audit a complete geographic transformation pipeline. Quantifies the perturbation introduced at each hop and reveals the shell game.

Usage

```
audit_transformation(
  baseline_data,
  zip_zcta_map,
  hud_crosswalk,
  county_fips,
  variable_name = "value",
  value_col = "estimate"
)
```

Arguments

baseline_data	Data frame with baseline data at source geography
zip_zcta_map	ZIP-ZCTA association crosswalk
hud_crosswalk	HUD ZIP-County crosswalk
county_fips	Target county FIPS code
variable_name	Name of the variable being tracked (for reporting)
value_col	Name of the value column in baseline_data

Value

An object of class "shellgame_audit" with audit results

Examples

```
## Not run:
result <- audit_transformation(
  baseline_data = hennepin_zcta_baseline,
  zip_zcta_map = hennepin_zip_zcta_map,
  hud_crosswalk = hennepin_hud_crosswalk,
  county_fips = "27053",
  variable_name = "population"
)
summary(result)

## End(Not run)
```

check_census_key	<i>Check for Census API key</i>
------------------	---------------------------------

Description

Validates that a Census API key is available for tidycensus.

Usage

```
check_census_key(install = FALSE)
```

Arguments

install Logical, whether to install the key for future sessions

Value

Invisible TRUE if key exists, stops with error if not

create_audit_report	<i>Create complete audit report</i>
---------------------	-------------------------------------

Description

Generates all visualizations for an audit.

Usage

```
create_audit_report(
  audit_result,
  zcta_baseline_sf = NULL,
  zcta_geometric_sf = NULL,
  county_sf = NULL
)
```

Arguments

audit_result A shellgame_audit object

zcta_baseline_sf
 Optional: SF object with baseline ZCTAs

zcta_geometric_sf
 Optional: SF object with geometric ZCTAs

county_sf Optional: SF object with county boundary

Value

List of ggplot2 objects

extract_perturbed_population
Extract perturbation by receiving county

Description

Returns a data frame of counties that received population redistributed from the target county during the transformation, ordered by magnitude.

Usage

```
extract_perturbed_population(audit_result, top_n = 10)
```

Arguments

audit_result A shellgame_audit object

top_n Number of top counties to return (default: 10)

Value

Data frame with columns: county, value

get_zcta_baseline	<i>Get ACS baseline data for ZCTAs</i>
-------------------	--

Description

Fetches ACS 5-year estimates for a specified variable at the ZCTA level using the Census API via the tidycensus package. Requires a Census API key (see https://api.census.gov/data/key_signup.html) and the tidycensus package to be installed.

Usage

```
get_zcta_baseline(variable, year = 2022, zctas = NULL)
```

Arguments

variable	ACS variable code (e.g., "B01001_001" for total population)
year	ACS year (default: 2022)
zctas	Optional character vector of ZCTAs to filter to

Value

Data frame with columns: zcta, estimate, moe

Examples

```
## Not run:  
# Get population for all ZCTAs  
pop_data <- get_zcta_baseline("B01001_001", year = 2022)  
  
# Get population for specific ZCTAs  
hennepin_zctas <- c("55401", "55402", "55403")  
pop_data <- get_zcta_baseline("B01001_001", zctas = hennepin_zctas)  
  
## End(Not run)
```

pad_geoid	<i>Pad GEOID to 5 digits</i>
-----------	------------------------------

Description

Ensures geographic identifiers are zero-padded to 5 digits.

Usage

```
pad_geoid(geoid)
```

Arguments

geoid Character or numeric vector of geographic identifiers

Value

Character vector of 5-digit zero-padded GEOIDs

Examples

```
pad_geoid(c("123", "45678", 789))
#> [1] "00123" "45678" "00789"
```

plot_baseline_zctas *Plot baseline ZCTAs*

Description

Creates a map showing the baseline ZCTAs used in the analysis.

Usage

```
plot_baseline_zctas(zcta_sf, county_sf, title = "Baseline ZCTAs")
```

Arguments

zcta_sf SF object with ZCTA geometries
county_sf SF object with county boundary
title Plot title

Value

A ggplot2 object

plot_geometric_vs_relationship
Plot geometric vs relationship membership

Description

Visualizes the discrepancy between geometric intersection and relationship-based membership.

Usage

```
plot_geometric_vs_relationship(  
  zcta_baseline_sf,  
  zcta_geometric_sf,  
  county_sf,  
  title = "Geometric vs Relationship Membership"  
)
```

Arguments

<code>zcta_baseline_sf</code>	SF object with baseline ZCTAs (relationship-based)
<code>zcta_geometric_sf</code>	SF object with all geometrically intersecting ZCTAs
<code>county_sf</code>	SF object with county boundary
<code>title</code>	Plot title

Value

A ggplot2 object

`plot_transformation_perturbation`
Plot transformation perturbation

Description

Creates a simple bar chart showing baseline vs recovered values.

Usage

```
plot_transformation_perturbation(audit_result)
```

Arguments

<code>audit_result</code>	A shellgame_audit object
---------------------------	--------------------------

Value

A ggplot2 object

prep_hud_crosswalk *Prepare HUD ZIP-County crosswalk data*

Description

Standardizes HUD crosswalk data with proper column names and formatting.

Usage

```
prep_hud_crosswalk(data, ratio_col = "TOT_RATIO")
```

Arguments

data	Raw HUD crosswalk data frame
ratio_col	Name of the ratio column to use (default: "TOT_RATIO")

Value

Data frame with standardized columns: zip, county, tot_ratio

Examples

```
## Not run:  
hud_raw <- read.csv("HUD_ZIP_COUNTY.csv")  
hud <- prep_hud_crosswalk(hud_raw)  
  
## End(Not run)
```

prep_zip_zcta *Prepare ZIP-ZCTA crosswalk data*

Description

Standardizes ZIP-ZCTA crosswalk data with proper column names and formatting.

Usage

```
prep_zip_zcta(data, zip_col = NULL, zcta_col = "zcta")
```

Arguments

data	Raw ZIP-ZCTA crosswalk data frame
zip_col	Name of the ZIP code column (default: "ZIP_CODE" or "zip")
zcta_col	Name of the ZCTA column (default: "zcta")

Value

Data frame with standardized columns: zcta, zip

Examples

```
## Not run:  
zip_zcta_raw <- read.csv("ZiptoZCTA-Table 1.csv")  
zip_zcta <- prep_zip_zcta(zip_zcta_raw)  
  
## End(Not run)
```

print.shellgame_audit *Print method for shellgame_audit*

Description

Print method for shellgame_audit

Usage

```
## S3 method for class 'shellgame_audit'  
print(x, ...)
```

Arguments

- x A shellgame_audit object
- ... Additional arguments (ignored)

run_full_transformation
Run full transformation pipeline

Description

Executes both hops: ZCTA → ZIP → County. Tracks the complete swap from observed to imputed data.

Usage

```
run_full_transformation(  
  baseline_data,  
  zip_zcta_map,  
  hud_crosswalk,  
  value_col = "estimate",  
  county_fips = NULL  
)
```

Arguments

baseline_data	Data frame with ZCTA-level baseline data
zip_zcta_map	ZIP-ZCTA association table
hud_crosswalk	HUD ZIP-County crosswalk
value_col	Name of value column in baseline_data
county_fips	Optional county FIPS to filter final result

Value

List with intermediate and final results

Examples

```
## Not run:
result <- run_full_transformation(
  baseline_data = zcta_pop,
  zip_zcta_map = zip_zcta,
  hud_crosswalk = hud,
  value_col = "pop",
  county_fips = "27053"
)

## End(Not run)
```

summary.shellgame_audit

Summary method for shellgame_audit

Description

Summary method for shellgame_audit

Usage

```
## S3 method for class 'shellgame_audit'
summary(object, ...)
```

Arguments

object	A shellgame_audit object
...	Additional arguments (ignored)

transform_zcta_to_zip *Transform ZCTA data to ZIP level*

Description

Performs the first hop: ZCTA → ZIP using association-based allocation. This is where the first swap occurs: observed data → imputed data.

Usage

```
transform_zcta_to_zip(baseline_data, zip_zcta_map, value_col = "estimate")
```

Arguments

baseline_data Data frame with columns: zcta, and a value column
zip_zcta_map Data frame with columns: zcta, zip
value_col Name of the value column in baseline_data (default: "estimate")

Value

Data frame with columns: zip, value (allocated to ZIP level)

Examples

```
## Not run:  
zip_data <- transform_zcta_to_zip(  
  baseline_data = zcta_pop,  
  zip_zcta_map = zip_zcta_assoc,  
  value_col = "pop"  
)  
  
## End(Not run)
```

transform_zip_to_county

Transform ZIP data to County level

Description

Performs the second hop: ZIP → County using HUD TOT_RATIO allocation. This is where the second swap occurs: further imputation via proxy.

Usage

```
transform_zip_to_county(zip_data, hud_crosswalk, county_fips = NULL)
```

Arguments

`zip_data` Data frame with columns: zip, value
`hud_crosswalk` Data frame with columns: zip, county, tot_ratio
`county_fips` Optional FIPS code to filter to specific county

Value

Data frame with columns: county, value (allocated to county level)

Examples

```
## Not run:  
county_data <- transform_zip_to_county(  
  zip_data = zip_pop,  
  hud_crosswalk = hud,  
  county_fips = "27053"  
)  
  
## End(Not run)
```

Index

`audit_transformation`, 2

`check_census_key`, 3
`create_audit_report`, 3

`extract_perturbed_population`, 4

`get_zcta_baseline`, 5

`pad_geoid`, 5
`plot_baseline_zctas`, 6
`plot_geometric_vs_relationship`, 6
`plot_transformation_perturbation`, 7
`prep_hud_crosswalk`, 8
`prep_zip_zcta`, 8
`print.shellgame_audit`, 9

`run_full_transformation`, 9

`summary.shellgame_audit`, 10

`transform_zcta_to_zip`, 11
`transform_zip_to_county`, 11