Package: griddy (via r-universe)

May 30, 2026

Title Geospatial Distribution Dynamics for Simple Features

Version 0.1.0

Description Tools for exploratory geospatial distribution dynamics with 'sf' objects and tidy data. Provides pooled and time-specific classification of longitudinal spatial values, classic discrete Markov transition matrices, spatial Markov matrices conditioned on spatial-lag classes, endpoint and adjacent rank mobility summaries, and 'ggplot2' visualizations. Methods follow Rey (2001) <doi:10.1111/j.1538-4632.2001.tb00444.x> and Rey et al. (2016) <doi:10.1007/s10109-016-0234-x>; design is inspired by the Python 'PySAL' 'giddy' package <https://pysal.org/giddy/>.

URL https://github.com/dshkol/griddy, https://dshkol.github.io/griddy/

BugReports https://github.com/dshkol/griddy/issues

License MIT + file LICENSE

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

Depends R (>= 4.2)

Imports dplyr, ggplot2, rlang, scales, sf, spdep, tibble, tidyr

Suggests cancensus, knitr, microbenchmark, pkgdown, rmarkdown, sfdep, spData, testthat (>= 3.0.0), tidycensus

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

NeedsCompilation no

Author Dmitry Shkolnik [aut, cre]

Maintainer Dmitry Shkolnik <shkolnikd@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-05-30 14:10:35 UTC

RemoteUrl https://github.com/cran/griddy

RemoteRef HEAD

RemoteSha f47889b52c38e58f6a9f3ca24f4d5a659ad91c6e

Contents

`class_intervals` *Class Intervals Used By A griddy Object*

Description

Class Intervals Used By A griddy Object

Usage

```
class_intervals(x, ...)
```

Arguments

`x`	A griddy result object.
`...`	Reserved for future methods.

Value

A tibble of class intervals when available.

Examples

```
panel <- data.frame(id = rep(1:3, each = 2), year = rep(2020:2021, 3), value = 1:6)
classes <- classify_dynamics(panel, id, year, value, k = 3)
class_intervals(classes)
```

`classify_dynamics` *Classify Longitudinal Values For Distribution Dynamics*

Description

Converts numeric values in a long panel into ordered classes suitable for transition analysis.

Usage

```
classify_dynamics(
  data,
  id,
  time,
  value,
  method = c("pooled_quantile", "time_quantile", "fixed", "existing"),
  k = 5,
  breaks = NULL,
  labels = NULL
)
```

Arguments

`data`	A data frame or `sf` object in long format.
`id`, `time`, `value`	Columns identifying spatial unit, time, and value.
`method`	Classification method. `"pooled_quantile"` uses one set of quantile breaks across all periods. `"time_quantile"` uses period-specific ranks. `"fixed"` uses user-supplied `breaks`. `"existing"` treats `value` as an already classified state.
`k`	Number of quantile classes.
`breaks`	Breaks for `method = "fixed"`.
`labels`	Optional class labels.

Value

A data frame or `sf` object with a `class` column and class `grd_classes`.

Examples

```
panel <- data.frame(
  id = rep(letters[1:4], each = 3),
  year = rep(2020:2022, times = 4),
  value = c(8, 9, 11, 10, 12, 13, 15, 14, 16, 20, 22, 25)
)

classes <- classify_dynamics(panel, id, year, value, k = 3)
classes
class_intervals(classes)
```

`lag_intervals` *Spatial Lag Intervals Used By A Spatial Markov Object*

Description

Spatial Lag Intervals Used By A Spatial Markov Object

Usage

```
lag_intervals(x)
```

Arguments

`x`	A `grd_spatial_markov` object.

Value

A tibble of spatial-lag class intervals.

Examples

```
panel <- data.frame(
  id = rep(1:4, each = 2),
  year = rep(2020:2021, times = 4),
  value = c(1, 2, 2, 3, 4, 3, 5, 6)
)
grid <- sf::st_sf(
  id = 1:4,
  geometry = sf::st_make_grid(
    sf::st_bbox(c(xmin = 0, ymin = 0, xmax = 2, ymax = 2)),
    n = c(2, 2)
  )
) |>
  dplyr::mutate(
    nb = sfdep::st_contiguity(geometry),
    wt = sfdep::st_weights(nb)
  )
spatial <- spatial_markov(panel, id, year, value, geometry = grid, k = 2)
lag_intervals(spatial)
```

markov_dynamics *Estimate Classic Markov Transition Dynamics*

Description

Estimates transition counts and probabilities between adjacent periods for a classified long panel.

Usage

```
markov_dynamics(classes, id, time, state = class)
```

Arguments

classes	A classified data frame from classify_dynamics() or any data frame with ID, time, and state columns.
id, time, state	Columns identifying spatial unit, time, and state.

Value

A grd_markov object.

Examples

```
panel <- data.frame(
  id = rep(letters[1:4], each = 3),
  year = rep(2020:2022, times = 4),
  value = c(8, 9, 11, 10, 12, 13, 15, 14, 16, 20, 22, 25)
)

classes <- classify_dynamics(panel, id, year, value, k = 3)
markov <- markov_dynamics(classes, id, year, class)

markov
transition_matrix(markov)
steady_state(markov)
```

plot_rank_mobility *Plot Rank Mobility*

Description

Plot Rank Mobility

Usage

```
plot_rank_mobility(x)
```

Arguments

x	A grd_rank_mobility object returned by rank_mobility().

Value

A ggplot object.

---

plot_spatial_markov *Plot Spatial Markov Matrices*

Description

Plot Spatial Markov Matrices

Usage

```
plot_spatial_markov(x)
```

Arguments

x	A grd_spatial_markov object.

Value

A ggplot object.

Examples

```
panel <- data.frame(
  id = rep(1:4, each = 2),
  year = rep(2020:2021, times = 4),
  value = c(1, 2, 2, 3, 4, 3, 5, 6)
)
grid <- sf::st_sf(
  id = 1:4,
  geometry = sf::st_make_grid(
    sf::st_bbox(c(xmin = 0, ymin = 0, xmax = 2, ymax = 2)),
    n = c(2, 2)
  )
) |>
  dplyr::mutate(
    nb = sfdep::st_contiguity(geometry),
    wt = sfdep::st_weights(nb)
  )
spatial <- spatial_markov(panel, id, year, value, geometry = grid, k = 2)
plot_spatial_markov(spatial)
```

plot_transition_matrix *Plot A Transition Matrix*

Description

Plot A Transition Matrix

Usage

```
plot_transition_matrix(x)
```

Arguments

x	A grd_markov object.

Value

A ggplot object.

Examples

```
panel <- data.frame(id = rep(1:4, each = 2), year = rep(2020:2021, 4), value = 1:8)
markov <- panel |>
  classify_dynamics(id, year, value, k = 2) |>
  markov_dynamics(id, year, class)

plot_transition_matrix(markov)
```

rank_mobility *Compute Rank Mobility*

Description

Computes simple rank changes for map-ready exploratory analysis.

Usage

```
rank_mobility(
  data,
  id,
  time,
  value,
  compare = c("endpoint", "adjacent"),
  descending = TRUE
)
```

Arguments

data	A long data frame or sf object.
id, time, value	Columns identifying spatial unit, time, and value.
compare	"endpoint" compares first and last observed periods per unit. "adjacent" compares adjacent periods.
descending	If TRUE, larger values receive better ranks.

Value

A grd_rank_mobility data frame or sf object.

Examples

```
panel <- data.frame(
  id = rep(letters[1:4], each = 3),
  year = rep(2020:2022, times = 4),
  value = c(8, 9, 11, 10, 12, 13, 15, 14, 16, 20, 22, 25)
)

rank_mobility(panel, id, year, value)
```

---

spatial_markov *Estimate Spatial Markov Transition Dynamics*

Description

Estimates transition matrices conditioned on the class of each unit's spatial lag at the start of the transition period.

Usage

```
spatial_markov(
  data,
  id,
  time,
  value,
  geometry = NULL,
  listw = NULL,
  nb = NULL,
  k = 5,
  lag_k = k,
  class_method = c("pooled_quantile", "time_quantile", "fixed"),
  zero.policy = TRUE
)
```

Arguments

data	A long data frame or sf object.
id, time, value	Columns identifying spatial unit, time, and value.
geometry	An sf tibble with one row per spatial unit (a single time slice's geography), carrying nb and wt list-columns produced by sfdep::st_contiguity() and sfdep::st_weights(). The preferred input.
listw	A row-standardized spdep listw object. Accepted for compatibility with prior workflows; use geometry for new code.
nb	A spdep neighbor list, used only when both geometry and listw are NULL. Converted with spdep::nb2listw(style = "W").
k	Number of value classes.
lag_k	Number of spatial-lag classes.
class_method	Value classification method passed to classify_dynamics().
zero.policy	Passed to spdep lag/weight helpers.

Value

A grd_spatial_markov object.

Examples

```
panel <- data.frame(
  id = rep(1:4, each = 3),
  year = rep(2020:2022, times = 4),
  value = c(1, 2, 3, 2, 3, 4, 4, 3, 5, 5, 6, 7)
)

grid <- sf::st_sf(
  id = 1:4,
  geometry = sf::st_make_grid(
    sf::st_bbox(c(xmin = 0, ymin = 0, xmax = 2, ymax = 2)),
    n = c(2, 2)
  )
) |>
  dplyr::mutate(
    nb = sfdep::st_contiguity(geometry),
    wt = sfdep::st_weights(nb)
  )

spatial <- spatial_markov(panel, id, year, value, geometry = grid, k = 2)

spatial
lag_intervals(spatial)
transition_matrix(spatial, "count", lag_class = "Q1")
```

steady_state *Estimate Stationary Distribution*

Description

Estimate Stationary Distribution

Usage

```
steady_state(x)
```

Arguments

x	A grd_markov object or transition probability matrix.

Value

A numeric vector.

Examples

```
prob <- matrix(c(0.8, 0.2, 0.3, 0.7), nrow = 2, byrow = TRUE)
steady_state(prob)
```

transition_matrix *Extract A Transition Matrix*

Description

Extract A Transition Matrix

Usage

```
transition_matrix(x, type = c("probability", "count"), lag_class = NULL)
```

Arguments

x	A grd_markov or grd_spatial_markov object.
type	"probability" or "count".
lag_class	Optional lag class for grd_spatial_markov.

Value

A matrix.

Examples

```
panel <- data.frame(
  id = rep(letters[1:4], each = 3),
  year = rep(2020:2022, times = 4),
  value = c(8, 9, 11, 10, 12, 13, 15, 14, 16, 20, 22, 25)
)
markov <- panel |>
  classify_dynamics(id, year, value, k = 3) |>
  markov_dynamics(id, year, class)

transition_matrix(markov, "count")
```

---

usjoin *US State Per-Capita Personal Income, 1929-2009*

Description

Per-capita personal income for the 48 contiguous US states in nominal dollars. Mirrors the canonical `usjoin` panel used in PySAL `giddy` and the spatial Markov literature, so examples and validation in this package are directly comparable to that reference work.

Usage

```
usjoin
```

Format

A tibble with 3,888 rows (48 states x 81 years) and 4 columns:

name State name.

state_fips State FIPS code, integer.

year Year, integer 1929 to 2009.

income Nominal per-capita personal income, integer USD.

Source

PySAL `libpysal` `examples/us_income/usjoin.csv`. The original series is constructed from US Bureau of Economic Analysis state personal income tables. See https://github.com/pysal/libpysal.

References

Rey, S. J. (2001). Spatial empirics for economic growth and convergence. *Geographical Analysis*, 33(3), 195-214.

Rey, S. J., Kang, W., & Wolf, L. (2016). The properties of tests for spatial effects in discrete Markov chain models of regional income distribution dynamics. *Journal of Geographical Systems*, 18(4), 377-398.

Examples

```
data(usjoin)
head(usjoin)
range(usjoin$year)
```

Index