Package: MyFirstFourStepModel (via r-universe)

May 30, 2026

Title Highly Simplified Four-Step Travel Demand Models for Teaching

Version 1.0.0

Description This package provides a highly simplified four-step model. Running the model requires only R, and is designed to be computationally efficient and run even on consumer-grade equipment.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports cli, dplyr, fs, ggplot2, glue, Hmisc, igraph, jsonlite, JuliaCall, magrittr, nnet, purrr, readr, readxl, scales, sf, stats, stringr, tibble, tidycensus, tidyr, tidyselect, tigris, usethis, writexl

URL http://projects.indicatrix.org/MyFirstFourStepModel/

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/rextendr/version 0.4.2

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0, xz

Depends R (>= 4.2)

VignetteBuilder knitr

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev git libglpk-dev make libgit2-dev libicu-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev xz-utils zlib1g-dev libclang-dev

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

Date/Publication 2026-03-13 21:03:00 UTC

RemoteUrl https://github.com/mattwigway/MyFirstFourStepModel

RemoteRef HEAD

RemoteSha 6b0cee89cf0a9303bb6e1cd625c51006f5d8ff36

Contents

`add_households` *Add households to particular Census tracts in the region*

Description

Add households to particular Census tracts in the region

Usage

```
add_households(scenario, tract, hhs)
```

Arguments

`scenario`	An existing land-use scenario
`tract`	a Census tract ID (or vector of tract IDs)-should be strings, not numbers
`hhs`	Households to add to the tract

Value

a copy of the scenario with the requested households added

The hhs parameter should be a table/tibble/data.frame of households with the following columns:

- income: Annual household income, dollars
- hhsize: Household size
- vehicles: Number of vehicles owned by the household
- workers: Number of workers in the household
- n: Number of households like this to add

If there are multiple tracts specified, the households will be evenly split among them

---

`calibrate_trip_distance_beta`

This calibrates the trip distance beta, using the method described in Merlin (2020) A new method using medians to calibrate single-parameter spatial interaction models, JTLU. The basic idea is that we adjust the beta for the decay function until half the distance-weighted accessibility occurs in the median travel distance. For NHB, productions and attractions will be the same

---

Description

This calibrates the trip distance beta, using the method described in Merlin (2020) A new method using medians to calibrate single-parameter spatial interaction models, JTLU. The basic idea is that we adjust the beta for the decay function until half the distance-weighted accessibility occurs in the median travel distance. For NHB, productions and attractions will be the same

Usage

```
calibrate_trip_distance_beta(productions, attractions, median_dist_km, dmat)
```

`calibrate_trip_distance_betas`

Calibrate betas for all trip types median_distances should be a list with element HBW, HBO, and NHB with median crow-flies trip distances in each.

Description

Calibrate betas for all trip types median_distances should be a list with element HBW, HBO, and NHB with median crow-flies trip distances in each.

Usage

```
calibrate_trip_distance_betas(balanced, marginals, median_distances_m)
```

`estimate` *Estimate a four step model for later use, based on 2017 NHTS data and PSRC household survey data (for distribution functions).*

Description

Load the NHTS data with load_nhts(), and if desired filter the households table to just the households you want to use in estimation. Julia must already be installed. If you get an error that Julia is not found, you need to set the JULIA_HOME environment variable to the directory containing the Julia executable.

Usage

```
estimate(
  nhts,
  osm,
  state,
  county,
  year,
 highway_types = c("motorway", "motorway_link", "trunk", "trunk_link", "primary",
    "primary_link")
)
```

Arguments

`nhts`	Path to 2017 NHTS CSV files
`osm`	Path to OSM .pbf file
`state`	State to estimate model for
`county`	County (or vector of counties) to estimate model for
`year`	Year of ACS and LODES data to use (if you get 404 errors, you are probably trying to use a year that LODES is not available for)
`highway_types`	OSM highway= tags to include in network, default "motorway", "motorway_link", "trunk", "trunk_link", "primary", "primary_link"

`estimate_vmt` *Calculates VMT based on flows.*

Description

Note that this is total VMT for the period the flows are for; you would have to do this for each period to get total daily VMT.

Usage

```
estimate_vmt(model, network, link_flows, period)
```

Arguments

model	The estimated model object
network	The network to use (should be the same one used in network_assignment()
link_flows	Estimated link flows from the network assignment function
period	Time period (AM Peak, Midday, PM Peak, or Overnight)

---

get_mode_shares *This function uses the output of mode_choice() to calculate mode shares.*

Description

This function uses the output of mode_choice() to calculate mode shares.

Usage

```
get_mode_shares(flows_by_mode)
```

Arguments

flows_by_mode	output of mode_choice()

Value

a data frame with columns for car, bike, walk, and transit.

---

load_landuse_scenario *Load a land use scenario in Excel format*

Description

Load a land use scenario in Excel format

Usage

```
load_landuse_scenario(filename)
```

Arguments

filename	file to read from

load_model *Load a model*

Description

Load a model

Usage

```
load_model(filename)
```

Arguments

filename	File name or URL to load model from

load_model_v0 *Load a model in model format 0 (used with pre-2026 releases)*

Description

Load a model in model format 0 (used with pre-2026 releases)

Usage

```
load_model_v0(filename)
```

Arguments

filename	Filename or URL for model

load_nhts *Load 2017 NHTS data, handling types appropriately*

Description

Load 2017 NHTS data, handling types appropriately

Usage

```
load_nhts(base_path)
```

Arguments

base_path	path to folder containing 2017 NHTS CSV files

map_congestion *Produce a congestion map*

Description

Produce a congestion map

Usage

```
map_congestion(model, network, flows)
```

Arguments

model	the model in use
network	the network scenario in use (e.g. model$networks$baseline)
flows	output of network_assignment()

map_trip_distribution *Map trip distribution from a single trip.*

Description

Produces a map showing where trips produced in origin_tract are attracted.

Usage

```
map_trip_distribution(model, flows, timeperiod, triptype, origin_tract)
```

Arguments

model	the model to use
flows	trip_distribuion() results
timeperiod	Time period to map (AM Peak, Midday, PM Peak, Overnight)
triptype	Trip type to map (HBO, HBW, NHB)
origin_tract	Tract to map productions of

`map_trip_generation` *Map trip generation.*

Description

Produces a map showing where trips are produced or attracted.

Usage

```
map_trip_generation(model, trip_counts, end, timeperiod, triptype)
```

Arguments

`model`	the model to use
`trip_counts`	`trip_generation()` results
`end`	"productions" or "attractions"
`timeperiod`	Time period to map (AM Peak, Midday, PM Peak, Overnight)
`triptype`	Trip type to map (HBO, HBW, NHB)

`mode_choice` *This runs the mode choice step of the model, and returns flows differentiated by mode for each trip type and time of day.*

Description

This runs the mode choice step of the model, and returns flows differentiated by mode for each trip type and time of day.

Usage

```
mode_choice(model, scenario, flows)
```

Arguments

`model`	The estimated model object
`scenario`	The scenario to use
`flows`	The output of trip_distribution

`modify_ways` *Modify network attributes of existing OSM ways. Note that this currently only updates them for modeling; visualizations and GIS exports are not changed.*

Description

Modify network attributes of existing OSM ways. Note that this currently only updates them for modeling; visualizations and GIS exports are not changed.

Usage

```
modify_ways(
  network,
  ways,
  lanes_per_direction = NULL,
  highway_type = NULL,
  maxspeed_kph = NULL
)
```

Arguments

`network`	Network to modify
`ways`	OSM way ID or character vector of way IDs. Since OSM IDs are 64 bit integers and R does not support 64 bit integers, these should be strings
`lanes_per_direction`	Number of lanes per direction to set on the way (optional)
`highway_type`	Highway type of the way. Currently motorway has higher capacity and all other values are treated as the same.
`maxspeed_kph`	Maximum speed on the way, in kilometers per hour

`network_assignment` *This runs the network assignment step of the model.*

Description

Network assignment is based on a Frank-Wolfe static traffic assignment algorithm. This returns a list of link-level flows. It applies peaking factors and average occupancy before doing assignment to convert period person-trips to hourly vehicle-trips.

Usage

```
network_assignment(model, scenario, network, mode_flows, period)
```

Arguments

model	The estimated model
scenario	The land-use scenario to use
network	The network scenario to use
mode_flows	Flows by mode and time of day, output of mode_choice function
period	Time period to assign, can be "AM Peak", "Midday", "PM Peak", or "Overnight".

---

network_to_gis *Convenience function to export a network scenario to a GIS file*

Description

Convenience function to export a network scenario to a GIS file

Usage

```
network_to_gis(network, file)
```

Arguments

network	the network to export, e.g. model$networks$baseline
file	where to save the file (file type determined by extension).

---

save_landuse_scenario *Save a land use scenario in Excel format*

Description

Save a land use scenario in Excel format

Usage

```
save_landuse_scenario(scenario, filename)
```

Arguments

scenario	scenario to save (often model$scenarios$baseline)
filename	file to save in

save_model *Save a model*

Description

Save a model

Usage

```
save_model(model, filename)
```

Arguments

model	model to save
filename	file to save to (canonical extension .mf4sm)

trip_distribution *This runs the trip distribution step of the model*

Description

Trip distribution is based on the relative locations of the tracts (in marginals), the betas estimated during the estimation phase of the model, and the total trip productions and attractions.

Usage

```
trip_distribution(model, scenario, balanced)
```

Arguments

model	the model object, loaded via load_model()
scenario	the scenario to use, often model$scenarios$baseline or a scenario created with add_households()
balanced	the balanced productions and attractions, returned by trip_generation()

`trip_generation` *This runs the trip generation step of the model.*

Description

Trip production is based on the marginal distributions for vehicles, workers, household size, and income. Trip attraction is based on job counts. Attractions are balanced to match productions before being returned.

Usage

```
trip_generation(model, scenario)
```

Arguments

`model`	the model object, loaded via `load_model()`
`scenario`	the scenario to use, often `model$scenarios$baseline` or a scenario created with `add_households()`

`write_lm` *Write the minimal information to be able to reconstruct enough of an lm to be able to do prediction.*

Description

Write the minimal information to be able to reconstruct enough of an lm to be able to do prediction.

Usage

```
write_lm(model, archive, name)
```

`write_mnl` *Write just enough of a multinomial logit model that we can deserialize and apply it*

Description

Write just enough of a multinomial logit model that we can deserialize and apply it

Usage

```
write_mnl(model, archive, name)
```

Index