

# Package: edgeTransport (via r-universe)

October 14, 2026

**Title** Prepare EDGE Transport Data for the REMIND model

**Version** 3.22.0

**Description** EDGE-T is a fork of the GCAM transport module <https://jgcri.github.io/gcam-doc/energy.html#transportation> with a high level of detail in its representation of technological and modal options. It is a partial equilibrium model with a nested multinomial logit structure and relies on the modified logit formulation. Most of the sources are not publicly available. PIK-internal users can find the sources in the distributed file system in the folder ``/p/projects/rd3mod/inputdata/sources/EDGE-Transport-Standalone``.

**Depends** R (>= 3.5.0), data.table (>= 1.11.0), mrtransport (>= 0.12.0)

**License** GPL-3

**URL** <https://github.com/pik-piam/edgeTransport>

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**Date** 2026-10-14

**Config/testthat/edition** 3

**Imports** rmndt, magclass, rootSolve, madrat, magrittr, gdx2, piamutils (>= 0.2.1), zoo, mrdrivers, reporttransport (>= 1.1.0)

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, covr, piamPlotComparison, renv

**Config/roxygen2/version** 8.1.0

**Config/pak/sysreqs**  
cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev git libglpk-dev make libharfbuzz-dev libgit2-dev libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

**Repository** <https://pik-piam.r-universe.dev>

**Date/Publication** 2026-10-14 09:34:22 UTC

**RemoteUrl** <https://github.com/pik-piam/edgeTransport>

**RemoteRef** HEAD

**RemoteSha** 897b743b2b8685d142677c8a69c26d79184a629b

## Contents

|                                                   |    |
|---------------------------------------------------|----|
| calcEdgeTransportSA . . . . .                     | 3  |
| calculateSharesDiscreteChoice . . . . .           | 4  |
| checkForNAsDups . . . . .                         | 5  |
| getFilterEntriesUnivocalName . . . . .            | 5  |
| iterativeEdgeTransport . . . . .                  | 6  |
| toolAdjustDemandCovid . . . . .                   | 6  |
| toolApplyICEbanOnPreferences . . . . .            | 7  |
| toolApplyMixedTimeRes . . . . .                   | 7  |
| toolApplyScenPrefTrends . . . . .                 | 8  |
| toolApplyScenSpecEnInt . . . . .                  | 9  |
| toolApplyScenSpecLoadFactor . . . . .             | 9  |
| toolBuildHelpers . . . . .                        | 10 |
| toolCalculateAnnuity . . . . .                    | 11 |
| toolCalculateFleetComposition . . . . .           | 11 |
| toolCalculateFS3share . . . . .                   | 12 |
| toolCalculateFVdemand . . . . .                   | 13 |
| toolCalculateInitialIncoCost . . . . .            | 14 |
| toolCalculateSharesDecisionTree . . . . .         | 15 |
| toolCalculateVehicleDepreciationFactors . . . . . | 15 |
| toolCalibratePreferences . . . . .                | 16 |
| toolCheckAllLevelsComplete . . . . .              | 17 |
| toolCombineCAPEXandOPEX . . . . .                 | 17 |
| toolCombineCAPEXandOPEXiterative . . . . .        | 18 |
| toolDemandRegression . . . . .                    | 19 |
| toolDeriveTimeRes . . . . .                       | 20 |
| toolDiscreteChoice . . . . .                      | 21 |
| toolEdgeTransportSA . . . . .                     | 22 |
| toolGetCommonParameters . . . . .                 | 23 |
| toolLoadDecisionTree . . . . .                    | 24 |
| toolLoadInputs . . . . .                          | 24 |
| toolLoadmrdriversData . . . . .                   | 25 |
| toolLoadmrtransportData . . . . .                 | 25 |
| toolLoadPackageData . . . . .                     | 26 |
| toolLoadRDSinputs . . . . .                       | 26 |
| toolLoadREMINDesDemand . . . . .                  | 27 |
| toolLoadREMINDfuelCosts . . . . .                 | 27 |
| toolLoadScenParIncoCost . . . . .                 | 28 |
| toolNormalizePreferences . . . . .                | 29 |
| toolOrderandCheck . . . . .                       | 29 |
| toolPrepareAndCalibrateModelInput . . . . .       | 30 |
| toolPrepareDataEndogenousCosts . . . . .          | 32 |
| toolPrepareScenInputData . . . . .                | 32 |
| toolReLoadInputs . . . . .                        | 33 |
| toolTranslateTransportScenario . . . . .          | 34 |
| toolTraverseDecisionTree . . . . .                | 35 |
| toolUpdateEndogenousCosts . . . . .               | 35 |

**Index**[37](#)


---

|                     |                                                                                   |
|---------------------|-----------------------------------------------------------------------------------|
| calcEdgeTransportSA | <i>Generate EDGE-Transport Input Data for the REMIND model, madrat interface.</i> |
|---------------------|-----------------------------------------------------------------------------------|

---

**Description**

calcEdgeTransportSA() is a wrapper for toolEdgeTransportSA() to make use of madrat caching.

**Usage**

```
calcEdgeTransportSA(
  SSPscen,
  transportPolScen,
  isICEban = c(FALSE, FALSE),
  demScen = c("default", "default"),
  startyear = 2021,
 .gdxPath = NULL,
  outputFolder = NULL,
  isStored = FALSE,
  isTransportReported = TRUE,
  isTransportExtendedReported = FALSE,
  isREMINDinputReported = FALSE,
  isAnalyticsReported = FALSE,
  testIterative = FALSE
)
```

**Arguments**

|                             |                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------|
| SSPscen                     | SSP or SDP scenario                                                                   |
| transportPolScen            | EDGE-T transport policy scenario                                                      |
| isICEban                    | optional enabling of ICE ban                                                          |
| demScen                     | Demand scenario, used to apply reduction factors on total demands from the regression |
| startyear                   | Year after which policy differentiation sets in                                       |
| .gdxPath                    | Path to a GDX file to load price signals from a REMIND run                            |
| outputFolder                | Path to folder for storing output data                                                |
| isStored                    | Optional saving of intermediate RDS files                                             |
| isTransportReported         | Optional transport reporting in MIF format                                            |
| isTransportExtendedReported | Optional extension of transport reporting providing more detailed variables           |
| isREMINDinputReported       | Optional reporting of REMIND input data                                               |

isAnalyticsReported

Optional reporting of analytics data (e.g. variables over iterations)

testIterative    development setting: make standalone and iterative scripts comparable, sets sectorESdemand = REMINDsectorESdemand and iterations = 1 (cost module)

### Value

Transport input data for REMIND

### Author(s)

Jarusch Muessel, Johanna Hoppe, Alex K. Hagen

---

calculateSharesDiscreteChoice

*Calculate shares based on discrete choice model.*

---

### Description

Function works for the use of generic preferences as well as for inconvenience costs. If no preferences are provided the function sets them to one which is equivalent to pure description by inconvenience costs.

### Usage

```
calculateSharesDiscreteChoice(totPrice, lambda, pref = NULL)
```

### Arguments

|          |                                                                      |
|----------|----------------------------------------------------------------------|
| totPrice | total price of an option in a branch of the decision tree            |
| lambda   | exponent that determines the price sensitivity of the decision model |
| pref     | optional use of generic preference factors                           |

### Value

share of option in a branch of the decision tree

### Author(s)

Johanna Hoppe

---

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| checkForNAsDups | <i>Check a data.table for NAs and duplicates and throw an error if needed</i> |
|-----------------|-------------------------------------------------------------------------------|

---

**Description**

Check a data.table for NAs and duplicates and throw an error if needed

**Usage**

```
checkForNAsDups(dt, varname, codePosition)
```

**Arguments**

|              |                                      |
|--------------|--------------------------------------|
| dt           | data.table to be checked             |
| varname      | name of the variable                 |
| codePosition | position in the code to find the bug |

**Author(s)**

Johanna Hoppe

---

|                              |                                                                         |
|------------------------------|-------------------------------------------------------------------------|
| getFilterEntriesUnivocalName | <i>List associated univocalNames for any entry of the decision tree</i> |
|------------------------------|-------------------------------------------------------------------------|

---

**Description**

List associated univocalNames for any entry of the decision tree

**Usage**

```
getFilterEntriesUnivocalName(categories, decisionTree)
```

**Arguments**

|              |                                                                          |
|--------------|--------------------------------------------------------------------------|
| categories   | vector of categories to filter                                           |
| decisionTree | decision tree that contains the univocalNames associated to the category |

**Value**

list of categories and their associated univocalNames e.g. trn\_pass as list entry containing a vector c("Compact Car", "HSR", ..)

**Author(s)**

Johanna Hoppe

---

iterativeEdgeTransport

*EDGE-Transport iterative*


---

### Description

Run in the REMIND output folder in between iterations

### Usage

```
iterativeEdgeTransport()
```

### Author(s)

Johanna Hoppe, Alex K. Hagen

---

toolAdjustDemandCovid *Function to adjust the sector ES demand on CES level to the covid decrease in 2020*


---

### Description

The adjustment is based on the relative differences of our FE model projections in 2020 to the FE IEA data averaged over five years. In addition, the deviation in 2020 is corrected for earlier differences between IEA data and model projections, such that only the decrease in deviation due to covid is regarded here. The final differences used for this correction are stored in inst/extdata/IEAdemandDifferencesCovid.csv For more context, the full data and calculations can be found in /p/projects/edget/adjustmentDataFiles/IEAdemandDifferencesCovid\_compHistoricREMIND.xlsx

### Usage

```
toolAdjustDemandCovid(demandData)
```

### Arguments

|            |                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------|
| demandData | the energy service demand on CES level: trn_aviation_intl, trn_shipping_intl, trn_freight, trn_pass |
|------------|-----------------------------------------------------------------------------------------------------|

### Details

The demand projections are constantly shifted by an absolute difference from 2020 onwards  
This function is applied in toolDemandRegression() directly after the regression calculation as a default

### Value

demandData adjusted for the decrease in demand related to covid

---

 toolApplyICEbanOnPreferences

*Apply ICE ban on vehicle types that feature preference factors*


---

### Description

Apply ICE ban on vehicle types that feature preference factors

### Usage

```
toolApplyICEbanOnPreferences(preferenceTab, helpers, ICEbanYears)
```

### Arguments

|               |                                                                      |
|---------------|----------------------------------------------------------------------|
| preferenceTab | data.table including preferences for all levels of the decision tree |
| helpers       | list of helpers                                                      |
| ICEbanYears   | sequence of years in which ICEban is applied                         |

### Value

Preferences in accordance to the ICE ban policy

### Author(s)

Johanna Hoppe, Alex K. Hagen

---

 toolApplyMixedTimeRes *toolApplyMixedTimeRes*


---

### Description

Applies two different temporal resolutions on a data.table object in the edgeTransport data structure and do a linear approximation for the highRes data that is not available

### Usage

```
toolApplyMixedTimeRes(data, helpers, idcols = NULL)
```

### Arguments

|         |                                                                                                                                                               |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data    | data.table containing data in in the edgeTransport data structure (at least featuring univocalName, period, value)                                            |
| helpers | list containing several helpers used throughout the model. It includes dtTimeRes, a data.table containing the temporal resolution for different univocalNames |
| idcols  | optional supply of idcols for using approx_dt                                                                                                                 |

**Value**

data.table

**Author(s)**

Johanna Hoppe

---

|                                                                     |
|---------------------------------------------------------------------|
| toolApplyScenPrefTrends                                             |
| <i>Apply scenario specific adjustments to the preference trends</i> |

---

**Description**

Apply scenario specific adjustments to the preference trends

**Usage**

```
toolApplyScenPrefTrends(  
  baselinePrefTrends,  
  scenParPrefTrends,  
  GDPpcMER,  
  allEqYear,  
  GDPcutoff,  
  helpers  
)
```

**Arguments**

|                    |                                                                                            |
|--------------------|--------------------------------------------------------------------------------------------|
| baselinePrefTrends | Baseline preference trends                                                                 |
| scenParPrefTrends  | Scenario parameters to be applied on the preference trends                                 |
| GDPpcMER           | Per capita GDP based on market exchange rate                                               |
| allEqYear          | Year after which scenario differentiation sets in                                          |
| GDPcutoff          | Threshold used to categorize countries into different mitigation groups based on their GDP |
| helpers            | List containing several helpers used throughout the model                                  |

**Value**

Scenario specific preference trends

**Author(s)**

Johanna Hoppe, Alex K. Hagen

---

`toolApplyScenSpecEnInt`*Apply demand scenario specific adjustments to the energy intensity*

---

**Description**

Apply demand scenario specific adjustments to the energy intensity

**Usage**

```
toolApplyScenSpecEnInt(enInt, scenParEnergyIntensity, allEqYear, helpers)
```

**Arguments**

|                                     |                                                                         |
|-------------------------------------|-------------------------------------------------------------------------|
| <code>enInt</code>                  | Energy intensity input data supplied by mrtransport                     |
| <code>scenParEnergyIntensity</code> | Transport policy scenario specific energy intensity improvement factors |
| <code>allEqYear</code>              | Year after which scenario differentiation sets in                       |
| <code>helpers</code>                | List with helpers                                                       |

**Value**

data.table with scenario specific energy intensity input data

**Author(s)**

Johanna Hoppe, Alex K. Hagen

---

`toolApplyScenSpecLoadFactor`*Apply demand scenario specific adjustments to the load Factor*

---

**Description**

Apply demand scenario specific adjustments to the load Factor

**Usage**

```
toolApplyScenSpecLoadFactor(loadFactor, scenParLoadFactor, allEqYear, helpers)
```

**Arguments**

|                                |                                                                        |
|--------------------------------|------------------------------------------------------------------------|
| <code>loadFactor</code>        | load factor input data supplied by mrtransport                         |
| <code>scenParLoadFactor</code> | Scenario specific parameters to be applied on the baseline load factor |
| <code>allEqYear</code>         | Year after which scenario differentiation sets in                      |
| <code>helpers</code>           | List with helpers                                                      |

**Value**

Scenario specific load factor

**Author(s)**

Johanna Hoppe, Alex K. Hagen

---

|                  |                                                                    |
|------------------|--------------------------------------------------------------------|
| toolBuildHelpers | <i>Assemble the list of helpers shared across the EDGE-T model</i> |
|------------------|--------------------------------------------------------------------|

---

**Description**

Derives filterEntries (associated univocalNames per category) and the mixed temporal resolution, then collects them together with the structural mappings into the helpers list used throughout the model. Shared by toolLoadInputs() (full input preparation) and toolReloadInputs() (reload from RDS in iterative mode) so that the two paths cannot drift apart.

**Usage**

```
toolBuildHelpers(decisionTree, timeResDataBase)
```

**Arguments**

|                 |                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------|
| decisionTree    | data.table of the full edgeTransport decision tree                                                       |
| timeResDataBase | data.table with univocalName and period used to derive the temporal resolution (see toolDeriveTimeRes()) |

**Value**

list of helpers used throughout the model

**Author(s)**

Alex K. Hagen

---

toolCalculateAnnuity    *Calculate annuity for different vehicle types*

---

**Description**

Calculate annuity for different vehicle types

**Usage**

```
toolCalculateAnnuity(annuityCalc, helpers)
```

**Arguments**

|             |                                                                  |
|-------------|------------------------------------------------------------------|
| annuityCalc | input data for interest rate and service Life from edgeTransport |
| helpers     | list with helpers                                                |

**Author(s)**

Johanna Hoppe

---

toolCalculateFleetComposition  
*Calculate composition of the fleet*

---

**Description**

This function calculates the fleet composition based on the energy service demand, the composition of new sales each year and the service life of vehicles. With the help of the annual mileage, the absolute numbers of vehicles is calculated. If the remaining stock is sufficient to meet the energy service demand, 10

**Usage**

```
toolCalculateFleetComposition(  
  ESdemandFVsalesLevel,  
  vehDepreciationFactors,  
  vehSalesAndModeShares,  
  annualMileage,  
  loadFactor,  
  helpers  
)
```

**Arguments**

|                        |                                                                                   |
|------------------------|-----------------------------------------------------------------------------------|
| ESdemandFVsalesLevel   | Energy service demand on fuel vehicle level for new sales                         |
| vehDepreciationFactors | Factors for the depreciation of vehicles in each year of the maximum service Life |
| vehSalesAndModeShares  | Vehicle sales and mode shares for the decision tree                               |
| annualMileage          | Annual kilometers driven by a vehicle                                             |
| loadFactor             | Persons or tons loaded on each vehicle                                            |
| helpers                | List of helpers                                                                   |

**Value**

list of fleet composition data

**Author(s)**

Johanna Hoppe

---

toolCalculateFS3share    *toolCalculateFS3share*

---

**Description**

Calculates fuel subsector L3 shares

**Usage**

```
toolCalculateFS3share(
  endoCostData,
  timesteps,
  timeValue,
  preferences,
  lambdas,
  helpers
)
```

**Arguments**

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| endoCostData | data.table containing all cost components on technology level               |
| timesteps    | years for which to calculate FS3 shares                                     |
| timeValue    | data.table containing mode specific time value costs based on speed and gdp |
| preferences  | data.table containing preference trends                                     |
| lambdas      | data.table containing exponents for discrete choice calculation             |
| helpers      | list containing helpers like mappings, decisionTree etc.                    |

**Value**

data.table containing all cost components on technology level and their respective FS3 shares

**Author(s)**

Johanna Hoppe

---

toolCalculateFVdemand *Calculate energy service demand on fuel vehicle level*

---

**Description**

Calculate energy service demand on fuel vehicle level

**Usage**

```
toolCalculateFVdemand(  
  sectorESdemand,  
  salesAndModeShares,  
  helpers,  
  histESdemand = NULL,  
  baseYear = NULL  
)
```

**Arguments**

|                    |                                                       |
|--------------------|-------------------------------------------------------|
| sectorESdemand     | energy service demand on CES node level               |
| salesAndModeShares | vehicle sales and mode shares                         |
| helpers            | List with helpers                                     |
| histESdemand       | historical energy service demand for iterative EDGE-T |
| baseYear           | base year of historical energy service demand         |

**Value**

data.table

**Author(s)**

Johanna Hoppe

---

`toolCalculateInitialIncoCost`

*Apply regional differences for inconvenience cost start values based on ICE cost differences*

---

**Description**

Apply regional differences for inconvenience cost start values based on ICE cost differences

**Usage**

```
toolCalculateInitialIncoCost(  
  combinedCost,  
  incoCostStartVal,  
  annuity,  
  loadFactor,  
  annualMileage,  
  helpers  
)
```

**Arguments**

|                               |                                                |
|-------------------------------|------------------------------------------------|
| <code>combinedCost</code>     | total cost of ownership                        |
| <code>incoCostStartVal</code> | start values for inconvenience costs           |
| <code>annuity</code>          | calculated annuity for different vehicle types |
| <code>loadFactor</code>       | load factor data                               |
| <code>annualMileage</code>    | annual mileage data                            |
| <code>helpers</code>          | list with helpers                              |

**Value**

data.table including initial inconvenience costs from 1990-2020 for LDV 4W US\$/(plt)km

**Author(s)**

Johanna Hoppe

---

`toolCalculateSharesDecisionTree`*Calculate shares for all levels of the decision tree for a variable on FV level*

---

**Description**

Calculate shares for all levels of the decision tree for a variable on FV level

**Usage**

```
toolCalculateSharesDecisionTree(dtVariable, helpers)
```

**Arguments**

|                         |                                                                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dtVariable</code> | data.table of a transport variable, e.g., energy service demand with data at the most detailed level of of the decision tree (FV = fuel-vehicle) |
| <code>helpers</code>    | list of helpers                                                                                                                                  |

**Value**

data.table including shares of the variable for all levels of the decision tree

**Author(s)**

Johanna Hoppe

---

`toolCalculateVehicleDepreciationFactors`*Calculate vehicle depreciation factors for fleet tracking*

---

**Description**

Calculate vehicle depreciation factors for fleet tracking

**Usage**

```
toolCalculateVehicleDepreciationFactors(annuityCalc, helpers)
```

**Arguments**

|                          |                              |
|--------------------------|------------------------------|
| <code>annuityCalc</code> | parameters for annualization |
| <code>helpers</code>     | list with helpers            |

**Value**

data.table including vehicle depreciation factors over service life#'

**Author(s)**

Johanna Hoppe

---

|                                                              |
|--------------------------------------------------------------|
| toolCalibratePreferences                                     |
| <i>Calibrate the logit share weights to historical data.</i> |

---

**Description**

Calibrate the logit share weights to historical data.

**Usage**

```
toolCalibratePreferences(  
  sharesToBeCalibrated,  
  combinedCosts,  
  timeValueCost,  
  lambdas,  
  helpers  
)
```

**Arguments**

- |                      |                                                                                      |
|----------------------|--------------------------------------------------------------------------------------|
| sharesToBeCalibrated | Shares at each level of the decision tree for which preferences are to be calibrated |
| combinedCosts        | Annualized total cost of ownership                                                   |
| timeValueCost        | Time value cost for passenger transport modes                                        |
| lambdas              | Exponents for discrete choice function                                               |
| helpers              | list with helpers                                                                    |

**Value**

data.table with calibrated historical preferences

---

```
toolCheckAllLevelsComplete
    toolCheckAllLevelsComplete
```

---

**Description**

Checks whether data is complete for all levels of decision tree

**Usage**

```
toolCheckAllLevelsComplete(data, decisionTree, name)
```

**Arguments**

|              |                                                                        |
|--------------|------------------------------------------------------------------------|
| data         | data.table containing data in all levels format that should be checked |
| decisionTree | data.table containing full edgeTransport decision Tree                 |
| name         | name of variable to be checked                                         |

**Value**

data.table

**Author(s)**

Johanna Hoppe

---

```
toolCombineCAPEXandOPEX
    Function that converts CAPEX and OPEX into US$/(p|t)km and provides them combined in a structured format
```

---

**Description**

Function that converts CAPEX and OPEX into US\$/(p|t)km and provides them combined in a structured format

**Usage**

```
toolCombineCAPEXandOPEX(
  CAPEXtrackedFleet,
  nonFuelOPEXtrackedFleet,
  CAPEXother,
  nonFuelOPEXother,
  fuelCosts,
  subsidies,
```

```
energyIntensity,  
loadFactor,  
annualMileage,  
annuity,  
helpers  
)
```

Arguments

|                         |                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------|
| CAPEXtrackedFleet       | CAPEX data for vehicle types that feature fleet tracking: Cars, trucks, busses         |
| nonFuelOPEXtrackedFleet | non-fuel OPEX data for vehicle types that feature fleet tracking: Cars, trucks, busses |
| CAPEXother              | CAPEX data for other vehicle types                                                     |
| nonFuelOPEXother        | non-fuel OPEX data for other vehicle types                                             |
| fuelCosts               | fuel cost data                                                                         |
| subsidies               | purchase price subsidy data                                                            |
| energyIntensity         | energy intensity data                                                                  |
| loadFactor              | load factor data                                                                       |
| annualMileage           | annual mileage data                                                                    |
| annuity                 | calculated annuity for different vehicle types                                         |
| helpers                 | list with helpers                                                                      |

Value

data.table including total costs of ownership in US\$/(plt)km

---

|                                  |                                                                                                                  |
|----------------------------------|------------------------------------------------------------------------------------------------------------------|
| toolCombineCAPEXandOPEXiterative | <i>Function that converts CAPEX and OPEX into US\$/(p t)km and provides them combined in a structured format</i> |
|----------------------------------|------------------------------------------------------------------------------------------------------------------|

---

Description

Function that converts CAPEX and OPEX into US\$/(plt)km and provides them combined in a structured format

Usage

```
toolCombineCAPEXandOPEXiterative(  
  CAPEXandNonFuelOPEX,  
  fuelCosts,  
  energyIntensity,  
  loadFactor,  
  annualMileage,  
  helpers  
)
```

Arguments

|                     |                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------|
| CAPEXandNonFuelOPEX | non-fuel OPEX and CAPEX data as combined in the previous EDGET run and loaded from RDS file |
| fuelCosts           | updated fuel cost from REMIND                                                               |
| energyIntensity     | energy intensity data                                                                       |
| loadFactor          | load factor data                                                                            |
| annualMileage       | annual mileage data                                                                         |
| helpers             | list with helpers                                                                           |

Value

data.table including total costs of ownership in US\$/(plt)km

---

|                      |                                         |
|----------------------|-----------------------------------------|
| toolDemandRegression | <i>Energy service demand projection</i> |
|----------------------|-----------------------------------------|

---

Description

Energy service demand projection

Usage

```
toolDemandRegression(  
  historicalESdemand,  
  GDPperCapitaPPP,  
  POP,  
  genParDemRegression,  
  scenParDemRegression,  
  scenParRegionalDemRegression,  
  scenParDemandFactors,  
  baseYear,  
  allEqYear,  
  helpers  
)
```

**Arguments**

|                              |                                                                |
|------------------------------|----------------------------------------------------------------|
| historicalESdemand           | Historical energy service demand                               |
| GDPperCapitaPPP              | GDP per capita based on purchase power parity                  |
| POP                          | Population data                                                |
| genParDemRegression          | General regression factors                                     |
| scenParDemRegression         | Scenario specific general regression factors                   |
| scenParRegionalDemRegression | Scenario specific regionally differentiated regression factors |
| scenParDemandFactors         | Demand scenario factors                                        |
| baseYear                     | End year of historical energy service demand data              |
| allEqYear                    | Start year of scenario differentiation                         |
| helpers                      | list with helpers                                              |

**Value**

Scenario specific energy service demand for all model years on CES level

**Author(s)**

Johanna Hoppe

---

|                   |                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------|
| toolDeriveTimeRes | <i>Derive the mixed temporal resolution used across EDGE-T from a reference data.table</i> |
|-------------------|--------------------------------------------------------------------------------------------|

---

**Description**

Vehicle types that feature fleet tracking are resolved annually (highRes), all other modes use a coarser resolution (lowTimeRes). The split is based on which univocalNames provide data for the full set of highRes periods. Shared by toolLoadInputs() and toolReLoadInputs() so the full-load and reload paths cannot drift apart.

**Usage**

```
toolDeriveTimeRes(dt)
```

**Arguments**

|    |                                                                                                                                                  |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------|
| dt | data.table containing at least univocalName and period columns (e.g. energy intensity data) that spans the full temporal resolution of the model |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------|

**Value**

list with dtTimeRes (unique univocalName/period combinations) and lowTimeRes (the coarse resolution periods)

**Author(s)**

Johanna Hoppe, Alex K. Hagen

---

|                    |                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------|
| toolDiscreteChoice | <i>Calculate vehicle sales shares and mode shares for all levels of the decisionTree.</i> |
|--------------------|-------------------------------------------------------------------------------------------|

---

**Description**

Function that traverses each level of the decision tree and calculates the shares according to the discrete choice approach

**Usage**

```
toolDiscreteChoice(input, lambdasDiscreteChoice, updatedEndoCosts, helpers)
```

**Arguments**

|                       |                                           |
|-----------------------|-------------------------------------------|
| input                 | dataset for discrete choice module        |
| lambdasDiscreteChoice | lambdas from the general model parameters |
| updatedEndoCosts      | updated endogenous costs                  |
| helpers               | list of helpers                           |

**Value**

calculated shares

**Author(s)**

Johanna Hoppe

---

|                     |                                                        |
|---------------------|--------------------------------------------------------|
| toolEdgeTransportSA | <i>Energy Demand Generator (EDGE)- Transport Model</i> |
|---------------------|--------------------------------------------------------|

---

### Description

The Edge Transport Model includes the transport specific input data preparation, a choice model to determine transport mode and technology shares, a demand regression and a fleet tracking for cars, busses and trucks

### Usage

```
toolEdgeTransportSA(
  SSPscen,
  transportPolScen,
  isICEban = c(FALSE, FALSE),
  demScen = c("default", "default"),
  startyear = 2021,
 .gdxPath = NULL,
  outputFolder = NULL,
  isStored = TRUE,
  isTransportReported = TRUE,
  isTransportExtendedReported = FALSE,
  isREMINDinputReported = FALSE,
  isAnalyticsReported = FALSE,
  testIterative = FALSE
)
```

### Arguments

|                             |                                                                                        |
|-----------------------------|----------------------------------------------------------------------------------------|
| SSPscen                     | SSP or SDP scenarios                                                                   |
| transportPolScen            | EDGE-T transport policy scenarios                                                      |
| isICEban                    | optional enabling of ICE ban                                                           |
| demScen                     | Demand scenarios, used to apply reduction factors on total demands from the regression |
| startyear                   | First time point in which policy differentiation sets in, cm_startyear in REMIND       |
| .gdxPath                    | Path to a GDX file to load price signals from a REMIND run                             |
| outputFolder                | Path to folder for storing output data                                                 |
| isStored                    | Optional saving of intermediate RDS files                                              |
| isTransportReported         | Optional transport reporting in MIF format                                             |
| isTransportExtendedReported | Optional extension of transport reporting providing more detailed variables            |
| isREMINDinputReported       | Optional reporting of REMIND input data                                                |

isAnalyticsReported

Optional reporting of analytics data (e.g. variables over iterations)

testIterative    development setting: make standalone and iterative scripts comparable, sets sectorESdemand = REMINDsectorESdemand and iterations = 1 (cost module)

### Value

Transport input data for REMIND

### Author(s)

Johanna Hoppe, Jarusch Müßel, Alois Dirnaichner, Marianna Rottoli, Alex K. Hagen

---

toolGetCommonParameters

*Distribute common parameters to standalone and iterative EDGET script*

---

### Description

Distribute common parameters to standalone and iterative EDGET script

### Usage

```
toolGetCommonParameters(startyear, ICEbanBefore = FALSE, ICEbanAfter = FALSE)
```

### Arguments

|              |                                                                                  |
|--------------|----------------------------------------------------------------------------------|
| startyear    | First time point in which policy differentiation sets in, cm_startyear in REMIND |
| ICEbanBefore | Boolean to determine if ICEban is in effect before startyear                     |
| ICEbanAfter  | Boolean to determine if ICEban is in effect from startyear on                    |

### Value

a list of parameters: allEqYear, ICEbanYears, GDPcutoff, baseYear, hybridElecShare

### Author(s)

Alex K. Hagen

---

|                      |                                                                                 |
|----------------------|---------------------------------------------------------------------------------|
| toolLoadDecisionTree | <i>Read and build the complete structure of the edgeTransport decision tree</i> |
|----------------------|---------------------------------------------------------------------------------|

---

**Description**

Read and build the complete structure of the edgeTransport decision tree

**Usage**

```
toolLoadDecisionTree(regionAggregation = "iso")
```

**Arguments**

regionAggregation  
choose one of the different options for regional aggregation (isolregionCode21|regionCode12)

**Value**

data.table of full spatially extended edgeTransport decision tree

**Author(s)**

Johanna Hoppe

---

|                |                                                           |
|----------------|-----------------------------------------------------------|
| toolLoadInputs | <i>Load all inputs that are required to run the model</i> |
|----------------|-----------------------------------------------------------|

---

**Description**

Load all inputs that are required to run the model

**Usage**

```
toolLoadInputs(SSPscen, transportPolScen, demScen, hybridElecShare, allEqYear)
```

**Arguments**

|                  |                                                                                              |
|------------------|----------------------------------------------------------------------------------------------|
| SSPscen          | SSP or SDP scenarios                                                                         |
| transportPolScen | EDGE-T transport policy scenarios                                                            |
| demScen          | Demand scenario, used to apply reduction factors on total demands from the regression        |
| hybridElecShare  | Share of electricity in Hybrid electric vehicles                                             |
| allEqYear        | Year after which scenario differentiation sets in, last year in which scenarios are the same |

**Value**

list with different input data sets

**Author(s)**

Johanna Hoppe, Alex K. Hagen

---

toolLoadmrdriversData *Load GDP and Population data from mrdrivers*

---

**Description**

Load GDP and Population data from mrdrivers

**Usage**

```
toolLoadmrdriversData(SSPscen, helpers, allEqYear)
```

**Arguments**

|           |                                                                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSPscen   | SSP scenarios for which the mrdrivers input data shall be loaded                                                                                              |
| helpers   | list containing several helpers used throughout the model. It includes dtTimeRes, a data.table containing the temporal resolution for different univocalNames |
| allEqYear | Year after which policy differentiation sets in                                                                                                               |

**Value**

list of data.tables containing mrdrivers input data

---

toolLoadmrtransportData  
*Load input data from the mrtransport package*

---

**Description**

Load input data from the mrtransport package

**Usage**

```
toolLoadmrtransportData(SSPscen = "SSP2")
```

**Arguments**

|         |                                                                    |
|---------|--------------------------------------------------------------------|
| SSPscen | SSP scenario for which the mrtransport input data should be loaded |
|---------|--------------------------------------------------------------------|

**Value**

list of data.tables with the mrtransport input data

---

|                     |                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------|
| toolLoadPackageData | <i>Load input data from the edgeTransport package and choose data according to SSP and transport policy scenario</i> |
|---------------------|----------------------------------------------------------------------------------------------------------------------|

---

### Description

Load input data from the edgeTransport package and choose data according to SSP and transport policy scenario

### Usage

```
toolLoadPackageData(SSPs, transportPolS, demScenario = NULL)
```

### Arguments

|               |                                                                        |
|---------------|------------------------------------------------------------------------|
| SSPs          | SSP scenarios for which the package data should be loaded              |
| transportPolS | transport policy scenarios for which the package data should be loaded |
| demScenario   | demand scenario for which the package input data should be loaded      |

### Value

list of data.tables with the package input data

---

|                   |                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| toolLoadRDSinputs | <i>Creates RDS files for iterative REMIND-EDGE runs from csv input files. Existing files are overwritten silently. Does not return anything.</i> |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Description

Creates RDS files for iterative REMIND-EDGE runs from csv input files. Existing files are overwritten silently. Does not return anything.

### Usage

```
toolLoadRDSinputs(edgeTransportFolder, inputFiles)
```

### Arguments

|                     |                                                                              |
|---------------------|------------------------------------------------------------------------------|
| edgeTransportFolder | directory in the output folder of the REMIND run that stores EDGET RDS files |
| inputFiles          | list of filename strings that should be read in                              |

---

 toolLoadREMINDesDemand

*Load REMIND energy service demand*


---

### Description

Load the energy service demand from a REMIND fulldata.gdx on sector level (CES level) in [bn (plt)km/yr] and the requested temporal resolution

### Usage

```
toolLoadREMINDesDemand(gdxPath, helpers)
```

### Arguments

|         |                             |
|---------|-----------------------------|
| gdxPath | path to REMIND fulldata.gdx |
| helpers | list of helpers             |

---

 toolLoadREMINDfuelCosts

*Load fuel prices from a REMIND fulldata.gdx in [US\$/MJ] and map them on to the edgeTransport decision tree. The output is provided in the same spatial resolution as the transferred gdx file and the temporal resolution is set according to the param yrs.*

---

### Description

Load fuel prices from a REMIND fulldata.gdx in [US\$/MJ] and map them on to the edgeTransport decision tree. The output is provided in the same spatial resolution as the transferred gdx file and the temporal resolution is set according to the param yrs.

### Usage

```
toolLoadREMINDfuelCosts(
  gdxPath,
  hybridElecShare,
  helpers,
  transportFolder = ".",
  iterationNumber = NULL
)
```

**Arguments**

|                 |                                                                         |
|-----------------|-------------------------------------------------------------------------|
| gdxPath         | path to REMIND fulldata.gdx                                             |
| hybridElecShare | Share of electricity in Hybrid electric vehicles                        |
| helpers         | list with helpers                                                       |
| transportFolder | folder to search for fuel prices from previous iterations for averaging |
| iterationNumber | iteration number to store fuel costs for later iterations               |

**Value**

fuel costs on technology level

---

toolLoadScenParIncoCost

*Function to load a parameter set of inconvenience start costs for a specific scenario(combination) from the csv file in the package*

---

**Description**

Function to load a parameter set of inconvenience start costs for a specific scenario(combination) from the csv file in the package

**Usage**

```
toolLoadScenParIncoCost(SSPs, transportPols)
```

**Arguments**

|               |                            |
|---------------|----------------------------|
| SSPs          | SSP scenarios              |
| transportPols | transport policy scenarios |

**Value**

list with different parameter sets, one for each scenario

**Author(s)**

Alex K. Hagen

---

`toolNormalizePreferences`*Normalize preferences so that the maximum in each branch of the decision tree equals 1*

---

**Description**

Normalize preferences so that the maximum in each branch of the decision tree equals 1

**Usage**

```
toolNormalizePreferences(preferenceTab)
```

**Arguments**

`preferenceTab` data.table including preferences for all levels of the decision tree

**Value**

Normalized preferences

**Author(s)**

Johanna Hoppe

---

`toolOrderandCheck`      *toolOrderandCheck*

---

**Description**

sort data.table according to edgeTransport data structure and check for NAs If checkCompletness is activated it is further checked, if the full data set is provided. In case of fleetVars == TRUE the full data set is reduced to the vehicle types that feature fleet tracking

**Usage**

```
toolOrderandCheck(  
  data,  
  decisionTree,  
  yrs = NULL,  
  checkCompletness = FALSE,  
  fleetVars = FALSE  
)
```

**Arguments**

|                   |                                                                                        |
|-------------------|----------------------------------------------------------------------------------------|
| data              | data.table containing data in all levels format that should be checked                 |
| decisionTree      | data.table containing full edgeTransport decision Tree                                 |
| yrs               | temporal resolution                                                                    |
| checkCompleteness | if activated it is further checked, if the full data set is provided                   |
| fleetVars         | if activated the full data set is reduced to vehicle types that feature fleet tracking |

**Value**

data.table

**Author(s)**

Johanna Hoppe

---

toolPrepareAndCalibrateModelInput

*Prepare scenario specific input data and calibrate historical preferences*

---

**Description**

Applies the scenario specific changes to the raw input data, calibrates the historical preferences from the historical energy service demand and assembles the preference trends (calibrated + scenario specific, mixed time resolution, optional ICE ban, normalized). Shared by toolEdgeTransportSA() (standalone) and iterativeEdgeTransport() (first call in a REMIND run) so that the two paths cannot drift apart.

**Usage**

```
toolPrepareAndCalibrateModelInput(
  genModelPar,
  scenModelPar,
  inputDataRaw,
  commonParams,
  isICEban,
  helpers
)
```

**Arguments**

|              |                                                                            |
|--------------|----------------------------------------------------------------------------|
| genModelPar  | General model parameters                                                   |
| scenModelPar | Transport scenario (SSPscen + demScen + polScen) specific model parameters |
| inputDataRaw | Raw input data, including the (deaggregated) REMINDfuelCosts               |

|              |                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------|
| commonParams | List of common parameters from toolGetCommonParameters(), supplying allEqYear, GDPcutoff and ICEbanYears |
| isICEban     | Vector of length two indicating an ICE ban before/after the startyear                                    |
| helpers      | List with helpers                                                                                        |

## Details

The returned inputData is complete for standalone mode. The iterative caller appends CAPEXandNonFuelOPEX, which is returned separately and is the only mode specific entry left. histPrefs is returned because both callers hand it on to storeData(), which writes it to 3\_Calibration; nothing else reads it. It is a calibration artefact rather than a scenario input: the calibration runs on historical periods only, and scenario differentiation starts after allEqYear (2020 at the earliest), so in standalone mode the result does not vary with the transport policy scenario. In iterative mode it does, but only through the REMIND fuel costs, which toolLoadREMINDfuelCosts() back-extrapolates from 2025 into the calibration years.

inputData is the interface that both producers of model input must satisfy: this function on the first call, and toolReloadInputs() on every later REMIND Nash iteration. That is why annualMileage, timeValueCosts and histESdemand travel through inputData instead of being read from inputDataRaw at the point of use - inputDataRaw does not exist on a reload iteration. Keep the members in step with the file list in toolReloadInputs().

What belongs in inputData, and what does not:

- scenario processed data consumed by the model (scenSpecPrefTrends, scenSpecLoadFactor, combinedCAPEXandOPEX, initialIncoCosts)
- raw data consumed unchanged by the model (annualMileage, timeValueCosts, histESdemand)
- scenSpecEnIntensity and upfrontCAPEXtrackedFleet, which no model function reads; they travel in inputData only so that storeData() writes them to 2\_InputDataPolicy
- CAPEXandNonFuelOPEX (iterative only), which is persisted to RDS so that the next REMIND Nash iteration can recombine it with the updated fuel costs

Raw inputs that scenario processing supersedes (energyIntensityRaw, loadFactorRaw, the CAPEX and OPEX tables, subsidies) and the driver data (GDP, population) stay out of inputData. They are read from inputDataRaw where they are needed - by toolPrepareScenInputData() inside this function and by toolDemandRegression() in the caller - and reach the output path from inputDataRaw.

Note on regional resolution: the function is agnostic to the number of regions. In iterative mode the REMIND fuel costs must already be deaggregated to 21 regions and appended to inputDataRaw before this function is called, since they enter the cost combination and thereby the preference calibration.

## Value

List with inputData, the calibrated historical preferences (histPrefs) and CAPEXandNonFuelOPEX, which only the iterative caller needs

## Author(s)

Alex K. Hagen

---

toolPrepareDataEndogenousCosts  
*Format all cost components*

---

**Description**

Format all cost components

**Usage**

```
toolPrepareDataEndogenousCosts(inputData, lambdas, helpers)
```

**Arguments**

|           |                                           |
|-----------|-------------------------------------------|
| inputData | List containing inputData                 |
| lambdas   | exponents for discrete choice calculation |
| helpers   | List with helpers                         |

**Value**

data.table including all cost components

**Author(s)**

Johanna Hoppe

---

toolPrepareScenInputData  
*Apply transport scenario specific adjustments to input data*

---

**Description**

Apply transport scenario specific adjustments to input data

**Usage**

```
toolPrepareScenInputData(  
  genModelPar,  
  scenModelPar,  
  inputDataRaw,  
  allEqYear,  
  GDPcutoff,  
  helpers  
)
```

**Arguments**

|              |                                                                            |
|--------------|----------------------------------------------------------------------------|
| genModelPar  | General model parameters                                                   |
| scenModelPar | Transport scenario (SSPscen + demScen + polScen) specific model parameters |
| inputDataRaw | Raw input data                                                             |
| allEqYear    | Year after which scenario differentiation sets in                          |
| GDPcutoff    | GDP cutoff to differentiate between regions                                |
| helpers      | List with helpers                                                          |

**Value**

List of data.tables with scenario specific input data

**Author(s)**

Johanna Hoppe

---

|                  |                                                                                                                                                         |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| toolReloadInputs | <i>called from iterativeEdgeTransport() reloads scenario specific input-Data from RDS files in EDGE-T folder instead of generation via mr-transport</i> |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Description**

called from iterativeEdgeTransport() reloads scenario specific inputData from RDS files in EDGE-T folder instead of generation via mrtransport

**Usage**

```
toolReloadInputs(edgeTransportFolder)
```

**Arguments**

|                     |                                                                              |
|---------------------|------------------------------------------------------------------------------|
| edgeTransportFolder | folder where the RDS files from last iterativeEdgeTransport() run are stored |
|---------------------|------------------------------------------------------------------------------|

**Value**

list with different input data sets

**Author(s)**

Alex K. Hagen

---

```
toolTranslateTransportScenario
```

*Function to rename scenarios between REMIND and EDGE-Transport naming conventions It is used at the interfaces REMIND-EDGETiterative, EDGETiterative-REMIND The same translations have to be done in mrremind readEDGETransport for REMIND input data generation*

---

### Description

Function to rename scenarios between REMIND and EDGE-Transport naming conventions It is used at the interfaces REMIND-EDGETiterative, EDGETiterative-REMIND The same translations have to be done in mrremind readEDGETransport for REMIND input data generation

### Usage

```
toolTranslateTransportScenario(
  demScen,
  transportPolScen,
  direction = c("EDGEtoREMIND", "REMINDtoEDGE")
)
```

### Arguments

|                  |                                                                                                                |
|------------------|----------------------------------------------------------------------------------------------------------------|
| demScen          | demand scenario, if demScen is in mapping it will be translated otherwise returned unchanged                   |
| transportPolScen | transport policy scenario if transportPolScen is in mapping it will be translated otherwise returned unchanged |
| direction        | one of two labels indicating the direction of the translation                                                  |

### Value

list of translated demScen and transportPolScen in the desired model name logic

### Author(s)

Alex K. Hagen

---

`toolTraverseDecisionTree`*Calculate data for the next higher level of the decision tree.*

---

**Description**

Function that aggregates the data for the next higher level of the decision tree

**Usage**

```
toolTraverseDecisionTree(data, upperLevel, decisionTree)
```

**Arguments**

|                           |                                                   |
|---------------------------|---------------------------------------------------|
| <code>data</code>         | data.table containing the data of the lower level |
| <code>upperLevel</code>   | name of the upper level in the decision tree      |
| <code>decisionTree</code> | full edgeTransport decision tree                  |

**Value**

data.table

**Author(s)**

Johanna Hoppe

---

`toolUpdateEndogenousCosts`*toolUpdateEndogenousCosts*

---

**Description**

Provides updates for endogenous cost components e.g. inconvenience costs for cars

**Usage**

```
toolUpdateEndogenousCosts(  
  dataEndoCosts,  
  depreciationFactors,  
  scenParIncoCost,  
  allEqYear,  
  timeValue,  
  preferences,  
  lambdas,  
  helpers,
```

```

    isICEban,
    ICEbanYears,
    vehiclesPerTech = NULL
)

```

### Arguments

|                     |                                                                                                                                                                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dataEndoCosts       | data.table containing all cost components for cars over the full range of policy years in a yearly resolution. Exogenous CAPEX and OPEX are provided over the full range. Endogenous cost components and FS3 shares are provided until 2020. Rest is filled with NA. |
| depreciationFactors | data.table containing vehicle depreciation factor for each year of service Life                                                                                                                                                                                      |
| scenParIncoCost     | data.table containing scenario specific parameters for inconvenience costs policy mask                                                                                                                                                                               |
| allEqYear           | year from which scenario-specific differentiation begins                                                                                                                                                                                                             |
| timeValue           | data.table containing mode specific time value costs based on speed and gdp                                                                                                                                                                                          |
| preferences         | preference factor trends                                                                                                                                                                                                                                             |
| lambdas             | data.table containing exponents for discrete choice calculation                                                                                                                                                                                                      |
| helpers             | list containing helpers like mappings, decisionTree etc.                                                                                                                                                                                                             |
| isICEban            | optional enabling of ICE ban                                                                                                                                                                                                                                         |
| ICEbanYears         | sequence of years in which ICEban is applied                                                                                                                                                                                                                         |
| vehiclesPerTech     | data.table containing total number of vehicles for all years and regions                                                                                                                                                                                             |

### Value

list containing data.table with endogenous cost components over the full time span and additional data.tables for model behavior analysis

### Author(s)

Johanna Hoppe

# Index

calcEdgeTransportSA, [3](#)  
calculateSharesDiscreteChoice, [4](#)  
checkForNAsDups, [5](#)  
  
getFilterEntriesUnivocalName, [5](#)  
  
iterativeEdgeTransport, [6](#)  
  
toolAdjustDemandCovid, [6](#)  
toolApplyICEbanOnPreferences, [7](#)  
toolApplyMixedTimeRes, [7](#)  
toolApplyScenPrefTrends, [8](#)  
toolApplyScenSpecEnInt, [9](#)  
toolApplyScenSpecLoadFactor, [9](#)  
toolBuildHelpers, [10](#)  
toolCalculateAnnuity, [11](#)  
toolCalculateFleetComposition, [11](#)  
toolCalculateFS3share, [12](#)  
toolCalculateFVdemand, [13](#)  
toolCalculateInitialIncoCost, [14](#)  
toolCalculateSharesDecisionTree, [15](#)  
toolCalculateVehicleDepreciationFactors, [15](#)  
  
toolCalibratePreferences, [16](#)  
toolCheckAllLevelsComplete, [17](#)  
toolCombineCAPEXandOPEX, [17](#)  
toolCombineCAPEXandOPEXiterative, [18](#)  
toolDemandRegression, [19](#)  
toolDeriveTimeRes, [20](#)  
toolDiscreteChoice, [21](#)  
toolEdgeTransportSA, [22](#)  
toolGetCommonParameters, [23](#)  
toolLoadDecisionTree, [24](#)  
toolLoadInputs, [24](#)  
toolLoadmrdriversData, [25](#)  
toolLoadmrtransportData, [25](#)  
toolLoadPackageData, [26](#)  
toolLoadRDSinputs, [26](#)  
toolLoadREMINDesDemand, [27](#)  
toolLoadREMINDfuelCosts, [27](#)  
  
toolLoadScenParIncoCost, [28](#)  
toolNormalizePreferences, [29](#)  
toolOrderandCheck, [29](#)  
toolPrepareAndCalibrateModelInput, [30](#)  
toolPrepareDataEndogenousCosts, [32](#)  
toolPrepareScenInputData, [32](#)  
toolReLoadInputs, [33](#)  
toolTranslateTransportScenario, [34](#)  
toolTraverseDecisionTree, [35](#)  
toolUpdateEndogenousCosts, [35](#)