Package 'cartools'

August 20, 2018

Type Package

Title Tools for Understanding Highway Performance

Version 0.1.0

Description Analytical tools are designed to help people understand the complex relationships associated with freeway performance and traffic breakdown. Emphasis is placed on: (1) Traffic noise or volatility; (2) Driver behavior and safety; and (3) Stochastic modeling, models that explain breakdown and performance.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

RoxygenNote 6.1.0

Imports animation, devtools, dplyr, gapminder, ggplot2, graphics, grDevices, knitr, rlist, rmarkdown, roxygen2, sde, shiny, stats, tidyverse, usethis, utils

Suggests

VignetteBuilder knitr

NeedsCompilation no

Author Paul Ossenbruggen [aut, cre]

Maintainer Paul Ossenbruggen <pjo@unh.edu>

Repository CRAN

Date/Publication 2018-08-20 09:00:03 UTC

R topics documented:

`acceldown` *The* `acceldown` *function estimates the relative locations of a vehicle downstream of a bottleneck.*

Description

The `acceldown` function estimates the relative locations of a vehicle downstream of a bottleneck.

Usage

```
acceldown(tstart, tend, umn, usd, xstart,  step)
```

Arguments

`tstart`	start time in seconds, a number
`tend`	end time, a number
`umn`	start speed (fps) for vehicle in lane 1, a number
`usd`	speed volatility for `umn`, a number
`xstart`	start location of the vehicle (feet), a number
`step`	size in seconds, a number

Value

`acceldown` uses a stochastic model to show the location of the vehicle after it merges at a bottleneck.

Examples

```
acceldown(8.75, 33.25, 47.59, 16.133, -70.36, 0.25)
```

accelpass *The* accelpass *function estimates the relative locations of two vehicles where one passes the other.*

Description

The accelpass function estimates the relative locations of two vehicles where one passes the other.

Usage

```
accelpass(tend, umn, usd, xstart, xfunnel, leff, lane, step)
```

Arguments

tend	end time for a simulation run, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	start location for vehicle in lane 1 (feet), a number
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
lane	number, a number
step	size in seconds, a number

Value

accelpass uses a stochastic model to show the locations of vehicle accelerating to pass another vehicle traveling side-by-side at the speed.

Examples

```
accelpass(30, 41, 11, -1000, -500, 14, 1, 2)
```

animate_rr animate_rr *draws the path around the ring-road using data* LF *from* rrtrials.

Description

animate_rr draws the path around the ring-road using data LF from rrtrials.

Usage

```
animate_rr(LF, leff)
```

Arguments

LF,	a matrix
leff,	effective vehicle length (feet), number

---

bmfree *The* bmfree *function returns a* tux *matrix of time* t*, speed* u *and location* x *values.*

Description

The bmfree function returns a tux matrix of time t, speed u and location x values.

Usage

```
bmfree(umn, usd, N, Tup)
```

Arguments

umn	mean speed (mph), a number
usd	standard deviation of umn, a number
N	number of time-steps, a number
Tup	upper time range in minutes, a number

---

bmfree2 *The* bmfree2 *function returns a* tux *matrix of time* t*, speed* u *(fps) and location* x *(feet).*

Description

The bmfree2 function returns a tux matrix of time t, speed u (fps) and location x (feet).

Usage

```
bmfree2(umn, usd, tstart, tend, xstart, step, type)
```

Arguments

umn	mean speed (mph), a number
usd	standard deviation of usd (mph), a number
tstart	simulation start time, a number
tend	simulation end time, a number
xstart	location (feet), a number
step	time-step size, a number
type	logical for plotting where TRUE creates a plot

---

brkcross0 — brkcross0 *estimates the time, speed and location that a vehicle crosses bottleneck location* x = *0.*

Description

brkcross0 estimates the time, speed and location that a vehicle crosses bottleneck location x = 0.

Usage

```
brkcross0(vehicle, df)
```

Arguments

vehicle,	a number
df,	a dataframe

---

brkdelay — brkdelay *is a wrapper function for* brktrials2.

Description

brkdelay is a wrapper function for brktrials2.

Usage

```
brkdelay(niter, tend, umn, usd, xstart, xfunnel, leff, lane, step)
```

Arguments

niter	number of iterations, a number
tend	end time for a simulation run, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	a vector of nveh start locations, (feet), a vector
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
lane	a vector of nveh numbers, a vector
step	size in seconds, a number

Value

brkdelay returns t-x trajectories of nveh vehicles at a bottleneck.

brkdelay3 brkdelay3 *is a wrapper function for* brktrials3.

Description

brkdelay3 is a wrapper function for brktrials3.

Usage

```
brkdelay3(niter, nveh, tstart, tend, umn, usd, xstart, xfunnel, leff, step)
```

Arguments

niter	number of iterations, a number
nveh	number of vehicles, a number
tstart	start time for a simulation run, a number
tend	end time for a simulation run, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	start location, (feet), a number
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
step	size in seconds, a number

Value

brkdelay3 returns t-x trajectories of nveh vehicles at a bottleneck.

---

brksummary *The function* brksummary *is a wrapper for* accelpass *and* passplot.

Description

The function brksummary is a wrapper for accelpass and passplot.

Usage

```
brksummary(k, tend, umn, usd, xstart, xfunnel, leff, lane, step)
```

Arguments

k	traffic density (vehicles per mile, vpm), a number
tend	end time, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	start location for vehicle in lane 1 (feet), a number
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
lane	number, a number
step	size in seconds, a number

Value

brksummary contains a table of information used for run plot

Examples

```
brksummary(50, 30, 41, 11, -1000, -500, 14, 0, 2)
```

---

brktrials2 — brktrials2 *produces* t-x *trajectories for lead and following vehicles at a bottleneck*

Description

brktrials2 produces t-x trajectories for lead and following vehicles at a bottleneck

Usage

```
brktrials2(tend, umn, usd, xstart, xfunnel, leff, lane, step, type)
```

Arguments

tend	end time for a simulation run, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	a vector of nveh start locations, (feet), a vector
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
lane	a vector of nveh numbers, a vector
step	size in seconds, a number
type	0 no plots, 1 prediction plot, 2 all plots, a number

Value

brktrials2 returns t-x trajectories of nveh vehicles at a bottleneck.

Examples

```
brktrials2(30, 41, 11, xstart, -500, 14, lane, 0.5, 1)
```

---

brktrials3 — brktrials3 *produces* t-x *trajectories for lead and following vehicles at a bottleneck*

Description

brktrials3 produces t-x trajectories for lead and following vehicles at a bottleneck

Usage

```
brktrials3(nveh, umn, usd, tstart, tend, xstart, step, type, leff, xfunnel)
```

Arguments

nveh	number of vehicles entering the bottleneck, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility (mph) for umn, a number
tstart	start time, (seconds), a number
tend	end time, (seconds), a number
xstart	start location, (feet), a number
step	size in seconds, a number
type	TRUE to create plots or FALSE otherwise, a logical
leff	vehicle length in feet, a number
xfunnel	upstream location of bottleneck taper, a number

Value

brktrials3 returns t-x trajectories of nveh vehicles at a bottleneck.

Examples

```
brktrials3(4, 68.4, 4.4, 0, 30, -700, 0.25, FALSE, 14, -500)
```

brktrials3setup — brktrials3setup *produces* t-x *trajectories for lead and following vehicles at a bottleneck*

Description

brktrials3setup produces t-x trajectories for lead and following vehicles at a bottleneck

Usage

```
brktrials3setup(nveh, umn, usd, tstart, tend, xstart, step, type, leff)
```

Arguments

nveh	number of vehicles entering the bottleneck, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
tstart	start time, (seconds), a number
tend	end time, (seconds), a number
xstart	start location, (feet), a number
step	size in seconds, a number
type	TRUE or FALSE, a logical
leff	vehicle length in feet, a number

Value

brktrials3setup returns t-x trajectories of nveh vehicles at a bottleneck.

Examples

```
brktrials3setup(5, 68.4, 4.4, 0, 30, -700, 0.25, FALSE, 14)
```

capacityplot — *The function* capacityplot *creates a* (k,Q*) *plot indicating the success rate of sustaining capacity* Q*

Description

The function capacityplot creates a (k,Q*) plot indicating the success rate of sustaining capacity Q*

Usage

```
capacityplot(QUKtdayX)
```

Arguments

QUKtdayX	a data frame

Examples

```
capacityplot(QUKtdayX)
```

---

cfanim *uses a stochastic model of speed to describe driver's ability to maintain a speed* u.

Description

cfanim uses a stochastic model of speed to describe driver's ability to maintain a speed u.

Usage

```
cfanim(umn, usd, tup, dt)
```

Arguments

umn	mean speed (mph), a number
usd	standard deviation of umn, a number
tup	upper time range, a number
dt	time-step, a number

Value

The cfanim function returns information for a single vehicle used to analyze the effects of traffic noise on car-following. The function uses a stochastic traffic noise model, a Brownian motion or Wiener W model.

---

crossconflict *The function* crossconflict *a graph showing driver conflict of two vehicles changing lanes.*

Description

The function crossconflict a graph showing driver conflict of two vehicles changing lanes.

Usage

```
crossconflict()
```

Examples

```
crossconflict()
```

decelmerge *The* decelmerge *function estimates the relative locations of two vehicles where one passes the other.*

Description

The decelmerge function estimates the relative locations of two vehicles where one passes the other.

Usage

```
decelmerge(tstart, tend, umn, usd, xstart, xfunnel, leff)
```

Arguments

tstart	start time, a number
tend	end time, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	start location for vehicle in lane 1 (feet), a number
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number

Value

decelmerge uses a deterministic model to show the locations of vehicle accelerating to pass another vehicle traveling side-by-side at the same speed.

Examples

```
decelmerge(0, 10, 41, 0, -1000, -500, 14)
```

dfmanager dfmanager *is a wrapper function for* brktrials2

Description

dfmanager is a wrapper function for brktrials2

Usage

```
dfmanager(nveh, tend, umn, usd, xstart, xfunnel, leff, lane, step)
```

Arguments

nveh	number of vehicles in the simulation, a number
tend	end time for a simulation run, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	a vector of nveh start locations, (feet), a vector
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
lane	a vector of nveh numbers, a vector
step	size in seconds, a number

Value

dfmanager returns t-x trajectories of nveh vehicles at a bottleneck.

---

f1 — *Space headway* h *between the lead and following vehicles at time* t. *Lead and following vehicles use* gbm *and constant speed models, respectively.*

Description

Space headway h between the lead and following vehicles at time t. Lead and following vehicles use gbm and constant speed models, respectively.

Usage

```
f1(xl0, ul0, theta, xf1, t, t1)
```

Arguments

xl0	location of lead vehicle at t = 0, a number
ul0	pre-breakdown speed of lead vehicle, a number
theta	a gbm model parameter, a number
xf1	location of following vehicle at t = t1, a number
t	time, a number
t1	deceleration time, a number

f2 *Headway* h *between the lead and following vehicles at time* t*. Lead and following vehicles use constant speed models, respectively.*

Description

Headway h between the lead and following vehicles at time t. Lead and following vehicles use constant speed models, respectively.

Usage

```
f2(xl0, ul0, ul3, theta, t3, xf1, t, t1)
```

Arguments

xl0	location of lead vehicle at time t = t0, a number
ul0	pre-breakdown speed of lead vehicle, a number
ul3	breakdown speed of lead vehicle, a number
theta	a gbm model parameter, a number
t3	transition time, a number
xf1	location of following vehicle at time t = t1, a number
t	time, a number
t1	deceleration time, a number

findt1 findt1 *returns the deceleration time* t1*, the location* xf1*, and speed* uf1 *of the following vehicle at time* t1*.*

Description

findt1 returns the deceleration time t1, the location xf1, and speed uf1 of the following vehicle at time t1.

Usage

```
findt1(xl0, xf0, ul0, uf0, theta, leff)
```

Arguments

xl0	lead vehicle location at time t0, a number.
xf0	following vehicle location at time t0, a number.
ul0	lead vehicle speed, a number.
uf0	following vehicle speed, a number.
theta	a gbm model parameter, a number.
leff	effective vehicle length, a number.

findt1ab *The* findt1ab *returns the time* tf1 *and the location* xf1 *and speed* uf1 *of the following vehicle when the driver begins to decelerate.*

Description

The findt1ab returns the time tf1 and the location xf1 and speed uf1 of the following vehicle when the driver begins to decelerate.

Usage

```
findt1ab(xl0, xf0, uf0, ul0, tseq, useq, t1, t2)
```

Arguments

xl0	lead vehicle location at time t0, a number
xf0	following vehicle location at time t0, a number
uf0	following vehicle speed at time t0, a number
ul0	lead vehicle speed at time t0, a number
tseq	time sequence of the lead vehicle l, a vector
useq	speed sequence of the lead vehicle l, a vector
t1	start time, a number
t2	end time, a number

findt2 findt2 *returns the time* t2 *and* xf2 *when the following vehicle driver recognizes the lead vehicle has decelerated.*

Description

findt2 returns the time t2 and xf2 when the following vehicle driver recognizes the lead vehicle has decelerated.

Usage

```
findt2(L, theta, t4)
```

Arguments

L	lead vehicle information, a matrix
theta	a gbm model parameter, a number
t4	time, a number

`findt2ab` — `t2` *is the intersection point of the following driver sight-line and lead vehicle* `x` *trajectory. Lead vehicle location* `xl2` *and speed* `ul2` *at the intersection point, respectively.*

Description

`t2` is the intersection point of the following driver sight-line and lead vehicle `x` trajectory. Lead vehicle location `xl2` and speed `ul2` at the intersection point, respectively.

Usage

```
findt2ab(tstart, tmid, tend, ulstart, ulmid, xlstart, xlmid, xfstart, uf, lty, lwd, col)
```

Arguments

`tstart`	start time, a number
`tmid`	intermediate time, a number
`tend`	end time, a number
`ulstart`	lead vehicle starting speed at `tstart`, a number
`ulmid`	lead vehicle end speed at `tmid`, a number
`xlstart`	lead vehicle location at `tstart`, a number
`xlmid`	lead vehicle location at `tmid`, a number
`xfstart`	following vehicle location at `tstart`, a number
`uf`	following vehicle speed, a number
`lty`	line type, a number
`lwd`	line width, a number
`col`	line color, a factor

`findt5` — *The time* `t5` *when the following vehicle reaches the bottleneck or location* `x` = *0.*

Description

The time `t5` when the following vehicle reaches the bottleneck or location `x` = 0.

Usage

```
findt5(xf1, xf3, uf1, uf3, theta, t, t1, t2, t3, t4)
```

Arguments

xf1	location of the following vehicle at t1, a number
xf3	location of the following vehicle at t3, a number
uf1	speed of the following vehicle at t1, a number
uf3	speed of the following vehicle at t3, a number
theta	a gbm model parameter of the following vehicle, a number
t	unknown time t5, a number
t1	deceleration time, a number
t2	time when the vehicle sights lead vehicle, a number
t3	time when the vehicle obtains breakdown speed u0, a number
t4	upper end of search range, a number

---

findt5ab — *The time* t5 *when the following vehicle reaches the bottleneck or location* x = *0.*

Description

The time t5 when the following vehicle reaches the bottleneck or location x = 0.

Usage

```
findt5ab(xf1, xf2, uf1, uf2, t1, t2, t4)
```

Arguments

xf1	location of the following vehicle at time t1, a number
xf2	location of the following vehicle at time t2, a number
uf1	speed of the following vehicle at time t1, a number
uf2	speed of the following vehicle at time t2, a number
t1	deceleration time, a number
t2	time when the vehicle sights lead vehicle, a number
t4	upper end of search range, a number

`formqueue` *The* `formqueue` *function is used to explain queuing upstream of a bottleneck.*

Description

The `formqueue` function is used to explain queuing upstream of a bottleneck.

Usage

```
formqueue(tstart, tend, umn, usd, xstart, xfunnel, leff)
```

Arguments

`tstart`	start time, a number
`tend`	end time, a number
`umn`	start speed (mph) for vehicle in lane 1, a number
`usd`	speed volatility for `umn`, a number
`xstart`	start location for vehicle in lane 1 (feet), a number
`xfunnel`	upstream location where the lane drop starts (feet), a number
`leff`	effective vehicle length(feet), a number

Value

`formqueue` uses a stochastic model to illustrate an "idealistic" situation, a so-called a "form queue merge."

Examples

```
formqueue(0, 10, 41, 11.6, -1000, -500, 14)
```

`freeflowpass` *The* `freeflowpass` *function estimates the relative locations of two vehicles where one passes the other.*

Description

The `freeflowpass` function estimates the relative locations of two vehicles where one passes the other.

Usage

```
freeflowpass(tstart, tend, umn, usd, xstart, xfunnel, leff)
```

Arguments

tstart	start time, a number
tend	end time, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	start location for vehicle in lane 1 (feet), a number
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number

Value

freeflowpass uses a deterministic model to show the locations of vehicle accelerating to pass another vehicle traveling side-by-side at the same speed.

Examples

```
freeflowpass(0, 10, 41, 0, -1000, -500, 14)
```

---

gbm *Estimate speed* u *at time* t *using a geometric Brownian motion model.*

Description

Estimate speed u at time t using a geometric Brownian motion model.

Usage

```
gbm(u0, theta, t)
```

Arguments

u0	initial speed (fps), a number
theta,	a number
t	time (seconds), a number

gbmtheta *Estimate* theta *of a geometric Brownian motion model.*

Description

Estimate theta of a geometric Brownian motion model.

Usage

```
gbmtheta(upre, u0, t3)
```

Arguments

upre	pre-breakdown speed, a number
u0	breakdown speed, a number
t3	time when u0 is initiated, a number

gbmx *Estimate distance* x *given a geometric Brownian motion model and time t.*

Description

Estimate distance x given a geometric Brownian motion model and time t.

Usage

```
gbmx(u0, theta, t)
```

Arguments

u0	initial speed, a number
theta	a gbm model estimate, a number.
t	time, a number

h0 *Space headway* h0 *at time* t = *0.*

Description

Space headway h0 at time t = 0.

Usage

```
h0
```

Format

h0 is the minimum headway of 5280/k0 and hsafe(u0) where u0 is the lead vehicle speed at t = 0.

h0 distance between lead and following vehicles, measured in feet.

h1 *Space headway* h1 *at time* t1*, the time the following vehicle initiates deceleration.*

Description

Space headway h1 at time t1, the time the following vehicle initiates deceleration.

Usage

```
h1
```

Format

h1 is the minimum headway of based on the following vehicle sight-line and hsafe(u1) where u1 is the lead vehicle speed at t = t1.

h1 headway between lead and following vehicles, measured in feet.

---

h2 *Space headway* h2 *at time* t2*, the time the following vehicle driver sees the lead vehicle has decelerated.*

Description

Space headway h2 at time t2, the time the following vehicle driver sees the lead vehicle has decelerated.

Usage

```
h2
```

Format

h2 is the headway hsafe(u2) where u2 is the lead vehicle speed at t = t2.

h2 headway between lead and following vehicles, measured in feet.

---

h3 *Space headway* h3 *at time* t3*, the time the lead vehicle has decelerated to breakdown speed* u0*.*

Description

Space headway h3 at time t3, the time the lead vehicle has decelerated to breakdown speed u0.

Usage

```
h3
```

Format

h3 is the headway hsafe(u3) where u3 = u0, the lead vehicle speed at t = t3.

h3 headway between lead and following vehicles, measured in feet.

---

hsafe — *Safe headway* h *between lead and following vehicles. Speed* u *of lead vehicle is measured in fps.*

Description

Safe headway h between lead and following vehicles. Speed u of lead vehicle is measured in fps.

Usage

```
hsafe(u, leff)
```

Arguments

u	speed in units of fps, a number
leff	effective vehicle length in feet, a number

Examples

```
hsafe(60, 14)
```

L — L *is a matrix containing information about the lead vehicle. The bottleneck is located at* x = 0.

Description

L is a matrix containing information about the lead vehicle. The bottleneck is located at x = 0.

Usage

```
L
```

Format

A matrix of 4 columns and 4 rows where the first through fourth rows contain time t, speed u, location x and safe headway h information:

t0 time when a lead vehicle begins to decelerate.

t1 NA for a lead vehicle.

t2 NA for a lead vehicle.

t3 time when the lead vehicle reaches a breakdown speed.

u0 lead vehicle speed at t0.

u1 NA for a lead vehicle.

u2 NA for a lead vehicle.

u3 lead vehicle breakdown speed.

x0 lead vehicle location at t0.

x1 NA for a lead vehicle.

x2 NA for a lead vehicle.

x3 lead vehicle location at t3.

h0 safe headway at t0.

h1 NA for a lead vehicle.

h2 NA for a lead vehicle.

h3 safe headway at t3.

---

lane — lane *is a vector that designates the lane drivers are traveling in the* brktrials *function.*

Description

lane is a input vector to brtrials. For example, the lanes may be numbered: 0 1 2 1 2.

Format

A vector used by brktrials

lane a lane number.

---

lanedf — lanedf *returns a vector of vehicles* veh *in* lane.

Description

lanedf returns a vector of vehicles veh in lane.

Usage

```
lanedf(pick, nveh, lane)
```

Arguments

pick	a lane number
nveh,	a number
lane,	a vector

leff — leff *is the effective length of a vehicle.*

Description

leff is the effective length of a vehicle.

Usage

```
leff
```

Format

leff is the length of a vehicle.

leff effective vehicle length, measured in feet.

lfmatrix — *The function* lfmatrix *displays* cflist*, the lists* L *and* Fw *information for lead and following vehicles, respectively.*

Description

The function lfmatrix displays cflist, the lists L and Fw information for lead and following vehicles, respectively.

Usage

```
lfmatrix(cflist, nveh)
```

Arguments

cflist	information produced from the run and trial functions, a list.
nveh	number of vehicles in the investigation or vehicle = 1, 2, ..., nveh, a number

Value

cflist returns two matrices consisting of speed u and location x data, which are in turn are used to estimate performance measures for the simulation.

`logisticmodel` *The function* `logisticmodel` *estimates the parameters of logistic regression model using a binomial formula and logit link function for data* `QUKtdayX`.

Description

The function `logisticmodel` estimates the parameters of logistic regression model using a binomial formula and logit link function for data `QUKtdayX`.

Usage

```
logisticmodel(QUKtdayX, type)
```

Arguments

`QUKtdayX`	a matrix
`type`	logical

Value

If `type` is TRUE, plot a density-failure probability plot is created and a data frame is returned. If If `type` is FALSE, a data frame is returned.

Examples

```
logisticmodel(QUKtdayX, TRUE)
```

`merge` `merge` *produces* `t-x` *trajectories for a pair of lead and following vehicles.*

Description

`merge` produces `t-x` trajectories for a pair of lead and following vehicles.

Usage

```
merge(df1,df2,leff,step,ylim)
```

Arguments

`df1`	leading vehicle, a matrix
`df2`	following vehicle, a matrix
`leff`	vehicle length, a number
`step`	time-step size, a number
`ylim`	for plot, a vector

Value

The merge, a wrapper function for bmfree2, xabparam and hsafe, returns a smooth hsafe rule t-x trajectory. for the following vehicle. The lead vehicle trajectory is not affected. Incidentally, inputs df1 and df2 were created with bmfree2 and used for testing.

---

merge3 — merge3 *produces* t-x *trajectories for a pair of lead and following vehicles for* brktrial3.

Description

merge3 produces t-x trajectories for a pair of lead and following vehicles for brktrial3.

Usage

```
merge3(i,df1,df2,leff,step,xfunnel,usd,ylim,type)
```

Arguments

i	vehicle, a number
df1	leading vehicle, a matrix
df2	following vehicle, a matrix
leff	vehicle length, a number
step	time-step size, a number
xfunnel	upstream location of bottleneck taper, a number
usd	speed volatility (mph) for umn, a number
ylim	for plot, a vector
type	TRUE to create plots or FALSE otherwise, a logical

Value

The merge3, a wrapper function for bmfree2, xabparam and hsafe, returns a smooth hsafe rule t-x trajectory. for the following vehicle. The lead vehicle trajectory is not affected.

mergedemo — mergedemo *produces* t-x *trajectories for a pair of lead and following vehicles.*

Description

mergedemo produces t-x trajectories for a pair of lead and following vehicles.

Usage

```
mergedemo(df1,df2,leff,step)
```

Arguments

df1	leading vehicle, a matrix
df2	following vehicle, a matrix
leff	vehicle length, a number
step	time-step size, a number

Value

The mergedemo, a wrapper function for bmfree2, xabparam and hsafe, returns a smooth hsafe rule t-x trajectory. for the following vehicle. The lead vehicle trajectory is not affected. Incidentally, inputs df1 and df2 were created with bmfree2 and used for testing.

nopass — nopass *produces* t, u *and* x *for lead and following vehicles downstream of a bottleneck*

Description

nopass produces t, u and x for lead and following vehicles downstream of a bottleneck

Usage

```
nopass(veh, nope, leff)
```

Arguments

veh	vehicle, a number
nope	is a data.frame of leading and following vehicles: time, speed and location data, a data frame
leff	effective vehicle length in feet, a number

Value

brktrials returns a data frame speed and location downstream of bottleneck.

passplot — *The function* passplot *plots the data from* brksummary, freeflowpass, decelmerge *data frames.*

Description

The function passplot plots the data from brksummary, freeflowpass, decelmerge data frames.

Usage

```
passplot(df, title)
```

Arguments

df	a matrix
title	a character string

plotcrossover — *A* ggplot2 *plot of crossover traffic data.*

Description

A ggplot2 plot of crossover traffic data.

Usage

```
plotcrossover(umn, leff)
```

Arguments

umn	average speed in mph, a number
leff	effective vehicle length in feet, a number

Examples

```
plotcrossover(67, 14)
```

plotdesire3 plotdesire3 *creates* t-x *desire-line trajectories.*

Description

plotdesire3 creates t-x desire-line trajectories.

Usage

```
plotdesire3(tuxv, xfunnel)
```

Arguments

tuxv	estimates from brktrials3, a matrix
xfunnel,	location of the vehicle, a vector

plotfollow *Produces a time-distance* t-x *trajectory for* vehicle i = 2.

Description

Produces a time-distance t-x trajectory for vehicle i = 2.

Usage

```
plotfollow(L, Fw, t4)
```

Arguments

L	a collection of variables that describe the relationship between a lead l and a following f vehicle, a matrix.
Fw	a collection of variables that describe the relationship between a lead l and a following f vehicle, a matrix.
t4	upper range of the plot, a number.

plotfollow3 plotfollow3 *adds a new trajectory to an existing graph.*

Description

plotfollow3 adds a new trajectory to an existing graph.

Usage

```
plotfollow3(cflist, vehicle, uf, kf, ub, leff, t4)
```

Arguments

cflist	contains information about the lead vehicle, a list
vehicle	a vehicle identifier, a number
uf	speed (mph) of the following vehicle at time t0, a number
kf	density (vehicles per mile, vpm) of the following vehicle at time t0, a number
ub	breakdown speed (mph) for this trajectory, a number
leff	effective vehicle length, a number.
t4	upper range of plot, a number

plotlead *Produces a time-distance* t-x *trajectory for the lead vehicle,* vehicle = 1.

Description

Produces a time-distance t-x trajectory for the lead vehicle, vehicle = 1.

Usage

```
plotlead(L, theta, leff, xlim, ylim)
```

Arguments

L,	a matrix
theta	a gbm model parameter, a number
leff	effective vehicle length (feet), a number
xlim	plot range, a vector
ylim	plot range, a vector

plotmerge3 plotmerge3 *creates* t-x *safe headway trajectories.*

Description

plotmerge3 creates t-x safe headway trajectories.

Usage

```
plotmerge3(tuxv.fix, vehorder, xfunnel)
```

Arguments

tuxv.fix	estimates from brktrials3, a matrix
vehorder	vehicle arrivals at x = 0, a vector
xfunnel,	location of the vehicle, a vector

plotoptimize plotoptimise *produces* t-x *trajectories for* nveh *drivers*

Description

plotoptimise produces t-x trajectories for nveh drivers

Usage

```
plotoptimize(df,xfunnel, type)
```

Arguments

df	a data frame from brktrials2
xfunnel	upstream location where the lane drop starts (feet), a number
type	2 to create a plot, a number

Value

plotoptimize returns t-x trajectories for nveh drivers, who wish to optimize their individual desires without constraint.

plottux — plottux *produces* t-u *and* t-x *trajectories for a lead and following vehicles on a ring road,*

Description

plottux produces t-u and t-x trajectories for a lead and following vehicles on a ring road,

Usage

```
plottux(umn, usd, N, T, k0, leff)
```

Arguments

umn	mean speed (mph), a number
usd	standard deviation of umn, a number
N	number of time-steps, a number
T	upper time range in seconds, a number
k0	traffic density (vehicles per mile, vpm), a number
leff	effective vehicle length, a number

plotupstream — plotupstream *produces* t-x *trajectories for lead and following vehicles at a bottleneck*

Description

plotupstream produces t-x trajectories for lead and following vehicles at a bottleneck

Usage

```
plotupstream(pick, lane, nveh, df, xfunnel, leff, type)
```

Arguments

pick	a lane number
lane,	a vector
nveh,	a number
df	a data frame from brktrials2
xfunnel	upstream location where the lane drop starts (feet), a number
leff	effective vehicle length(feet), a number
type	2 to create a plot, a number

Value

plotupstream returns t-x trajectories of nveh vehicles at a bottleneck. Wrapper for vehdf, nopass and vehfix

---

QUKtdayX *I-93 data set*

Description

A data frame with 19044 observations on the following 6 variables.

Usage

```
data("QUKtdayX")
```

Format

A data frame with 19044 observations on the following 6 variables.

q a numeric vector

u a numeric vector

k a numeric vector

day a numeric vector

t a numeric vector

X a numeric vector

---

rrheadway rrheadway *is a function to estimate the traffic flow* q *on a ring road.*

Description

rrheadway is a function to estimate the traffic flow q on a ring road.

Usage

```
rrheadway(LF, dt)
```

Arguments

LF	location x output from rrtrials, a matrix.
dt	time-step, a number.

Value

The rrheadway uses the output matrix LF from rrtrials.

`rrheadwayu` `rrheadwayu` *is a function used to find the speed* `u` *on a ring road at* `x = 0`.

Description

`rrheadwayu` is a function used to find the speed `u` on a ring road at `x = 0`.

Usage

```
rrheadwayu(LFu, t5)
```

Arguments

`LFu`	speed `u` output from `rrtrials`, a matrix.
`t5`	time when vehicle crosses the line `x = 0`, a vector.

Value

The `rrheadwayu` uses the output matrix `LFu` from `rrtrials` and codet5.

`rrtrials` `rrtrials` *produces* `t-x` *trajectories for lead and following vehicles on a ring road*

Description

`rrtrials` produces `t-x` trajectories for lead and following vehicles on a ring road

Usage

```
rrtrials(umn, usd, k0, N, T, leff, nveh, xlim, ylim)
```

Arguments

`umn`	lead vehicle speed (mph), a number
`usd`	standard deviation of lead vehicle speed (mph), a number
`k0`	traffic density (vehicles per mile, vpm), a number
`N`	number of time-steps, a number
`T`	upper time range (seconds), a number
`leff`	effective vehicle length (feet), a number
`nveh`	number of following vehicles, a number
`xlim`	plot range along the `t` axis, a vector
`ylim`	plot range along the `x` axis, a vector

Value

The rrtrials function returns t-x trajectories. for the lead and following vehicles on a single-lane road. This is a strict car-following model. Crossing trajectories are disallowed. In addition, a following vehicle driver maintains a safe headway while attempting to minimize travel time. In other words, the following vehicle's movement is constrained. Output from The rrtrials is an output matrix called LF.

Examples

```
rrtrials(41, 11, 50, 120, 120, 14, 9,   c(0, 120), c(-1000, 7500))
rrtrials(2,  2, 55, 60, 60, 14,    9,   c(0, 60), c(-2000, 600))
rrtrials(18.8, 3.8, 55, 120, 120, 14, 9, c(0, 120), c(-1000, 2000))
```

---

run *The* run *function is a wrapper for the* trial *and* plotfollow3 *functions.*

Description

The run function is a wrapper for the trial and plotfollow3 functions.

Usage

```
run(nveh, ulead, ulead.sd, ubrkdown, ubrkdown.sd, k, k.sd, d, leff, xlim,
  ylim)
```

Arguments

nveh	number of vehicles in the investigation or vehicle = 1, 2, ..., nveh, a number
ulead	lead vehicle speed (mph) at time t0, i.e., before breakdown, a number
ulead.sd	standard deviation of ulead, a number
ubrkdown	breakdown speed (mph) of lead vehicle, a number
ubrkdown.sd	standard deviation of ubrkdown, a number
k	traffic density (vehicles per mile, vpm) of the pre-breakdown state, a number
k.sd	standard deviation of following vehicle breakdown speed, a number
d	deceleration rate of the lead l vehicle, a number
leff	effective vehicle length (feet), a number
xlim	plot range along the t axis, a vector
ylim	plot range along the x axis, a vector

Value

The function returns an t-x image and eight lists: output[[1]] returns a vector of run input values; output[[2]] returns a matrix of pre-breakdown ulead speed, breakdown ubrkdown speed, pre-breakdown k density values for each vehicle; output[[3]] returns a time matrix consisting of t0,t1,t2,t3,t4,t5; output[[4]], output[[5]], output[[6]] and output[[7]] returns matrices of speed u, location x and safe headway h values for each vehicle at times t0,t1,t2,t3, respectively. Finally, output[[8]] returns a list the cflist list consisting of information stored as L and Fw matrices.

Examples

```
run(10, 63, 10, 0,  0,  55, 5, 15, 14, c(0, 20),c(-1000, 200))
run(10, 63, 10, 34, 0.1,  55, 5, 15, 14, c(0, 60),c(-2000, 4000))
run(10, 63, 10, 10,  1,  55, 5, 15, 14, c(0, 40),c(-2000, 2500))
```

---

safecrossover *The function* safecrossover *a graph showing driver conflict of two vehicles changing lanes.*

Description

The function safecrossover a graph showing driver conflict of two vehicles changing lanes.

Usage

```
safecrossover(umn, leff)
```

Arguments

umn	average speed in mph, a number
leff	effective vehicle length in feet, a number

Examples

```
safecrossover(67, 14)
```

schematic *The function* schematic *defines key features of a simple bottleneck*

Description

The function schematic defines key features of a simple bottleneck

Usage

```
schematic()
```

Examples

```
schematic()
```

speedboxplot *The function* speedboxplot *plots the data from a data frame* QUKtdayX

Description

The function speedboxplot plots the data from a data frame QUKtdayX

Usage

```
speedboxplot(QUKtdayX)
```

Arguments

QUKtdayX,	a matrix

speedplot *A scatter plot of traffic data.*

Description

A scatter plot of traffic data.

Usage

```
speedplot(QUKtdayX)
```

Arguments

QUKtdayX,	a matrix

Examples

```
speedplot(QUKtdayX)
```

`speedts` `speedts` *speed data used by* `speedtsplot`

Description

`speedts` speed data for lead and following vehicles.

Usage

```
data("speedts")
```

Format

A data frame with observations on the following 3 variables.

`day` a numeric vector

`t` a numeric vector

`u` a numeric vector

`speedtsplot` *The function* `speedtsplot` *plots the data from the* `speedts` *data frame.*

Description

The function `speedtsplot` plots the data from the `speedts` data frame.

Usage

```
speedtsplot(speedts)
```

Arguments

`speedts,`	a matrix

`t0` *t0 is the time the analysis is initiated.*

Description

`t0` is the time the analysis is initiated.

Usage

```
t0
```

Format

t0 time, measured in seconds.

`t1` *t1 is time the following vehicle realizes the lead vehicle is decelerating.*

Description

`t1` is time the following vehicle realizes the lead vehicle is decelerating.

Usage

```
t1
```

Format

The following vehicle initiates deceleration.

t1 time, measured in seconds.

`t2` *`t2` is time the following vehicle realizes the lead vehicle is decelerating.*

Description

`t2` is time the following vehicle realizes the lead vehicle is decelerating.

Usage

```
t2
```

Format

The following vehicle sight-line intersects the lead vehicle trajectory.

t2 time, measured in seconds.

`t3` *`t3` is time the lead vehicle completes deceleration.*

Description

`t3` is time the lead vehicle completes deceleration.

Usage

```
t3
```

Format

The lead vehicle is traveling at `u3 = u0`, the breakdown speed at time `t3`.

t3 time, measured in seconds.

theta — *The exponential deceleration rate* theta *of the lead vehicle.*

Description

The exponential deceleration rate theta of the lead vehicle.

Usage

```
theta
```

Format

theta is the decelerate from speed upre to u0 over time t0 to t3.

theta lead vehicle deceleration rate, measured in feet per second.

trajectoryab — trajectoryab *fits a second-order velocity model and produces a* t-x *trajectory for* vehicle = i *where* i = 3,4, ..., nveh.

Description

trajectoryab fits a second-order velocity model and produces a t-x trajectory for vehicle = i where i = 3,4, ..., nveh.

Usage

```
trajectoryab(tstart, tend, ustart, uend, xstart, xend, step)
```

Arguments

tstart	time, a number
tend	time, a number
ustart	speed, a number
uend	speed, a number
xstart	location, a number
xend	location, a number
step	line type, a number

`trajectoryab3` `trajectoryab3` *fits a second-order velocity model and produces a* `t-x` *trajectory for* `vehicle = i` *where* `i = 3,4, ..., nveh`.

Description

`trajectoryab3` fits a second-order velocity model and produces a `t-x` trajectory for `vehicle = i` where `i = 3,4, ..., nveh`.

Usage

```
trajectoryab3(tstart, tend, ustart, uend, xstart, xend, step,type)
```

Arguments

`tstart`	time, a number
`tend`	time, a number
`ustart`	speed, a number
`uend`	speed, a number
`xstart`	location, a number
`xend`	location, a number
`step`	line type, a number
`type`	TRUE to create plots or FALSE otherwise, a logical

`trial` *The function* `trial` *produces traffic performance information, stored in lists* `L` *and* `Fw` *for lead and following vehicles, respectively/* `L` *and* `Fw` *contain the information needed to graph a time-distance* `t-x` *trajectories for two vehicles.*

Description

The function `trial` produces traffic performance information, stored in lists `L` and `Fw` for lead and following vehicles, respectively/ `L` and `Fw` contain the information needed to graph a time-distance `t-x` trajectories for two vehicles.

Usage

```
trial(ul, ubk, k, uf, d, leff, xlim, ylim)
```

Arguments

ul	pre-breakdown speed of lead l vehicle, a number
ubk	breakdown speed of lead l vehicle, a number
k	traffic density of the pre-breakdown state, a number
uf	pre-breakdown speed of following f vehicle, a number
d	deceleration rate of the first of lead l vehicle, a number
leff	effective vehicle length (feet), a number
xlim	plot range along the t axis, a vector
ylim	plot range along the x axis, a vector

Value

LF consists of a list of four outputs: (1) an input vector where speed is given in miles per hour (mph), where vehicle density is given in vehicles per mile, vpm, where deceleration rate is given in feet per second squared (fps2), and where vehicle length in feet; (2) an input vector using feet and second scales exclusively; a car-following matrix L; and (4) a output vector consisting various performance estimates of delay including traffic flow in the congested state and shock-wave speed for evaluating queuing.

---

tuxvfix3 — tuxvfix3 *fixes* hsafe *violations for vehicle* veh *in matrix* tuxv.

Description

tuxvfix3 fixes hsafe violations for vehicle veh in matrix tuxv.

Usage

```
tuxvfix3(i, vehorder, nveh, tuxv, ufix, xfix)
```

Arguments

i	index, a number
vehorder	vehicle, a vector
nveh	number of vehicle in df, a number
tuxv	a data frame created inbk2trials, a matrix
ufix	new speeds, a vector
xfix	new locations, a vector

uab *Estimate distance* u *using second-order speed function.*

Description

Estimate distance u using second-order speed function.

Usage

```
uab(u0,a,b,t,t0)
```

Arguments

u0	speed at t0, a number
a	second-order speed function parameter, a number
b	second-order speed function parameter, a number
t	forecast time, a number
t0	initial time t0, a number

Examples

```
uab(0,16,0,10,0)
```

ubreakdown *The* ubreakdown *function generates a breakdown speed vector* ubk *assuming a lognormal distribution.*

Description

The ubreakdown function generates a breakdown speed vector ubk assuming a lognormal distribution.

Usage

```
ubreakdown(nveh, ubk, ubk.sd)
```

Arguments

nveh	number of vehicles in the investigation or vehicle = 1, 2, ..., nveh, a number
ubk	average breakdown speed, a number
ubk.sd	standard deviation of ubk, a number

uf0 *The following* f *vehicle speed* uf0 *at time* t0.

Description

The following f vehicle speed uf0 at time t0.

Usage

```
uf0
```

Format

The following vehicle speed at t0 is assumed to be independent of the lead vehicle speed ul0.

uf0 speed, measured in feet per second.

uf1 *The following* f *vehicle speed* uf1 *at time* t1.

Description

The following f vehicle speed uf1 at time t1.

Usage

```
uf1
```

Format

The following vehicle speed at t1 is assumed to be dependent on the lead vehicle speed ul1.

uf1 speed, measured in feet per second.

uf2 *The following* f *vehicle speed* uf2 *at time* t2.

Description

The following f vehicle speed uf2 at time t2.

Usage

```
uf2
```

Format

The following vehicle speed at t2 is assumed to be dependent on the lead vehicle speed ul2.

uf2 speed, measured in feet per second.

uf3 *The following* f *vehicle speed* uf3 *at time* t0.

Description

The following f vehicle speed uf3 at time t0.

Usage

```
uf3
```

Format

The following vehicle speed at t3 is assumed to be equal to the breakdown speed u0.

uf3 speed, measured in feet per second.

uk_sdk *A speed volatility model.*

Description

A speed volatility model.

Usage

```
uk_sdk(QUKtdayX, type)
```

Arguments

QUKtdayX,	a matrix
type	logical

Value

If type is TRUE, plot a density-speed plot is created and a data frame is returned. If If type is FALSE, a data frame is returned.

Examples

```
uk_sdk(QUKtdayX, TRUE)
```

ul0 *The lead* l *vehicle speed* ul0 *at time* t0.

Description

The lead l vehicle speed ul0 at time t0.

Usage

```
ul0
```

Format

The lead vehicle speed at t0 is assumed to be independent of the pre-breakdown speed upre.

ul0 speed, measured in feet per second.

ul1 *The following* l *vehicle speed* ul1 *at time* t1.

Description

The following l vehicle speed ul1 at time t1.

Usage

```
ul1
```

Format

The lead vehicle speed at t1 is estimated with the exponential speed function gbm(upre, theta, t1).

ul1 speed, measured in feet per second.

ul2 *The following* l *vehicle speed* ul2 *at time* t2.

Description

The following l vehicle speed ul2 at time t2.

Usage

```
ul2
```

Format

The lead vehicle speed at t2 is estimated with the exponential speed function gbm(upre, theta, t2).

ul2 speed, measured in feet per second.

ul3 *The following* l *vehicle speed* ul3 *at time* t3.

Description

The following `l` vehicle speed `ul3` at time `t3`.

Usage

```
ul3
```

Format

The lead vehicle speed at `t2` is estimated with the exponential speed function `gbm(upre, theta, t3)`.

ul3 speed, measured in feet per second.

usdBBridge *The* usdBBridge *function from Iacus.*

Description

The `usdBBridge` function from Iacus.

Usage

```
usdBBridge(umn, usd, N, T)
```

Arguments

`umn`	mean speed (mph), a number
`usd`	standard deviation of `umn`, a number
`N`	number of time-steps, a number
`T`	upper time range in minutes, a number

Value

The `usdBBridge` function returns information for a single vehicle used to analyze the effects of traffic noise on car-following. The model, a stochastic traffic noise model, is assumed to be a Brownian bridge model. Distance `x` is estimated with the forward Euler method.

Examples

```
usdBBridge(41, 11, 90000, 15)
usdBBridge(18.8, 3.9, 90000, 15)
```

usdzipper — *The* usdzipper *function is used to perform a sensitivity analysis of a single vehicles in traffic breakdown.*

Description

The usdzipper function is used to perform a sensitivity analysis of a single vehicles in traffic breakdown.

Usage

```
usdzipper(tstart, tend, umn, usd, xstart, xend)
```

Arguments

tstart	start time, a number
tend	end time, a number
umn	start speed (mph) for vehicle in lane 1, a number
usd	speed volatility for umn, a number
xstart	start location for vehicle in lane 1 (feet), a number
xend	end location for vehicle in lane 1 (feet), a number

Value

usdzipper uses a deterministic model to illustrate an "idealistic" situation, a so-called a "zipper merge."

Examples

```
usdzipper(0, 5, 41, 11.6, 0, 500)
```

vehdf — vehdf *selects information about vehicle* veh *of the* df *matrix for analysis with the* brktrials2 *function.*

Description

vehdf selects information about vehicle veh of the df matrix for analysis with the brktrials2 function.

Usage

```
vehdf(veh, nveh, df)
```

Arguments

veh	vehicle, a number
nveh	number of vehicle in df, a number
df	a data frame created inbk2trials, a data frame

---

vehfix — vehfix *fixes* hsafe *violation with speed* ufix *and location* xfix *and* yfix *vectors from* brktrials2

Description

vehfix fixes hsafe violation with speed ufix and location xfix and yfix vectors from brktrials2

Usage

```
vehfix(veh, nveh, ufix, xfix, yfix, df)
```

Arguments

veh	vehicle, a number
nveh	number of vehicle in df, a number
ufix	speed, a vector
xfix	location, a vector
yfix	location, a vector
df	a data frame created in bk2trials, a data frame

---

vehid — vehid *selects information about vehicle* veh *from the* tuxv *matrix for analysis with the* brktrials3 *function.*

Description

vehid selects information about vehicle veh from the tuxv matrix for analysis with the brktrials3 function.

Usage

```
vehid(veh, tuxv)
```

Arguments

veh	vehicle, a number
tuxv	a data frame created inbrktrials3, a matrix

vehtfix vehtfix *fixes* hsafe *violations for vehicle* veh.

Description

vehtfix fixes hsafe violations for vehicle veh.

Usage

```
vehtfix(veh, nveh, df, tstart, tend, step, ufix, xfix)
```

Arguments

veh	vehicle, a number
nveh	number of vehicle in df, a number
df	a data frame created inbk2trials, a data frame
tstart	time, a number
tend	time, a number
step	size in seconds, a number
ufix	new speeds, a vector
xfix	new locations, a vector

xab *Estimate distance* x *using second-order speed function.*

Description

Estimate distance x using second-order speed function.

Usage

```
xab(x0,u0,a,b,t,t0)
```

Arguments

x0	location at t0, a number
u0	speed at t0, a number
a	second-order speed function parameter, a number
b	second-order speed function parameter, a number
t	forecast time, a number
t0	initial time t0, a number

Examples

```
xab(0,0,16,0,10,0)
```

xabmerge — xabmerge *produces* t-x *trajectories for a pair of lead and following vehicles.*

Description

xabmerge produces t-x trajectories for a pair of lead and following vehicles.

Usage

```
xabmerge(df1, df2, leff, step, k, type)
```

Arguments

df1	leading vehicle, a matrix
df2	following vehicle, a matrix
leff	vehicle length, a number
step	time-step size, a number
k	counter, a number
type	logical for plotting where TRUE creates a plot

Value

The xabmerge, a wrapper function for bmfree2, xabparam and hsafe, returns a smooth hsafe rule t-x trajectory. for the following vehicle. The lead vehicle trajectory is not affected. Incidentally, inputs df1 and df2 were created with bmfree2 and used for testing.

xabmerge3 — xabmerge3 *produces* t-x *trajectories for a pair of lead and following vehicles.*

Description

xabmerge3 produces t-x trajectories for a pair of lead and following vehicles.

Usage

```
xabmerge3(df1, df2, leff, step, k, type)
```

Arguments

df1	leading vehicle, a matrix
df2	following vehicle, a matrix
leff	vehicle length, a number
step	time-step size, a number
k	counter, a number
type	logical for plotting where TRUE creates a plot

Value

The xabmerge3, a wrapper function for bmfree2, xabparam and hsafe, returns a smooth hsafe rule t-x trajectory. for the following vehicle. The lead vehicle trajectory is not affected. Incidentally, inputs df1 and df2 were created with bmfree2 and used for testing.

---

xabparam *Estimate the parameters* a *and* b *of a first-order acceleration model* a + b * t.

Description

Estimate the parameters a and b of a first-order acceleration model a + b * t.

Usage

```
xabparam(tstart, tend, ustart, uend, xstart, xend)
```

Arguments

tstart	time in seconds, a number
tend	time in seconds, a number
ustart	speed, a number
uend	speed, a number
xstart	location, a number
xend	location, a number

Examples

```
xabparam(0, 18.59797, 92.4, 0, 0, 1397.045)
```

xf0 *The lead* f *vehicle location* xf0 *at time* t0.

Description

The lead f vehicle location xf0 at time t0.

Usage

```
xf0
```

Format

The following vehicle location at t0 is assumed to be dependent space headway h0.

xf0 distance, measured in feet.

xf1 *The lead* f *vehicle location* xf1 *at time* t1.

Description

The lead f vehicle location xf1 at time t1.

Usage

```
xf1
```

Format

The following vehicle location at t1 is assumed to be dependent space headway h1.

xf1 distance, measured in feet.

xf2 *The lead* f *vehicle location* xf2 *at time* t2.

Description

The lead f vehicle location xf2 at time t2.

Usage

```
xf2
```

Format

The following vehicle location at t2 is assumed to be dependent space headway h2.

xf2 distance, measured in feet.

xf3 *The lead* f *vehicle location* xf3 *at time* t3.

Description

The lead f vehicle location xf3 at time t3.

Usage

```
xf3
```

Format

The following vehicle location at t3 is assumed to be dependent space headway h3.

xf3 distance, measured in feet.

xfollow *Estimate* x *distance traveled using a constant speed model.*

Description

Estimate x distance traveled using a constant speed model.

Usage

```
xfollow(x0, u, t, t0)
```

Arguments

x0	starting location at time t0, a number
u	speed, a number
t	travel time, a number
t0	deceleration time, a number

xl0 *The lead* l *vehicle location* xl0 *at time* t0*.*

Description

The lead l vehicle location xl0 at time t0.

Usage

```
xl0
```

Format

The following vehicle location at t0 is assumed to be dependent space headway h0.

xl0 distance, measured in feet.

`xl1` *The lead* `l` *vehicle location* `xl1` *at time* `t1`.

Description

The lead `l` vehicle location `x1` at time `t1`.

Usage

```
xl1
```

Format

The lead vehicle location at `t1` is assumed to be dependent space headway `h1`.

xl1 distance, measured in feet.

`xl2` *The lead* `l` *vehicle location* `xl2` *at time* `t2`.

Description

The lead `l` vehicle location `xl2` at time `t2`.

Usage

```
xl2
```

Format

The lead vehicle location at `t2` is assumed to be dependent space headway `h2`.

xf2 distance, measured in feet.

xl3 *The lead* l *vehicle location* xl3 *at time* t3.

Description

The lead l vehicle location xl3 at time t3.

Usage

```
x3
```

Format

The lead vehicle location at t3 is assumed to be dependent space headway h3.

xl3 distance, measured in feet.

xstart *is a vector that designates the start* x *location for the* brktrials *function.*

Description

xstart is a input vector to brtrials. For example, the start locations of the vehicles may be: -900 -990 -1000 -1090 -1100 feet.

Format

A vector used by brktrials

xstart xstart is measured in feet

zipper *The* zipper *function can be used to simulate interaction among three vehicles in car following on a single lane or three vehicles merging on a two-lane highway.*

Description

The zipper function can be used to simulate interaction among three vehicles in car following on a single lane or three vehicles merging on a two-lane highway.

Usage

```
zipper(tstart, tend,
    ustart1, uend1, xstart1, xend1,
    ustart2, uend2, xstart2, xend2,
    ustart3, uend3, xstart3, xend3)
```

Arguments

tstart	start time, a number
tend	end time, a number
ustart1	start speed (mph) for vehicle in lane 1, a number
uend1	end speed (mph) for vehicle in lane 1, a number
xstart1	start location for vehicle in lane 1 (feet), a number
xend1	end location for vehicle in lane 1 (feet), a number
ustart2	start speed (mph) for vehicle in lane 2, a number
uend2	end speed (mph) for vehicle in lane 2, a number
xstart2	start location for vehicle in lane 2 (feet), a number
xend2	end location for vehicle in lane 2 (feet), a number
ustart3	start speed (mph) for vehicle in lane 3, a number
uend3	end speed (mph) for vehicle in lane 3, a number
xstart3	start location for vehicle in lane 3 (feet), a number
xend3	end location for vehicle in lane 3 (feet), a number

Value

`zipper` uses a deterministic model and animation to illustrate an "idealistic" situation, a so-called a "zipper merge."

Examples

```
zipper(0,60,90,90,0,5000,90,90,-200,4500,90,90,-500, 4000)
zipper(0,40,85,90,0,4000,90,99,0,4500,90,90,-500, 4200)
zipper(0,5,60,20,0,500,65,20,-100,467,80,20,-350,433)
```

Index