Package ‘mousetrack’

February 20, 2015

Type Package

Title Mouse-Tracking Measures from Trajectory Data

Version 1.0.0

Date 2015-01-29

Author Moreno I. Coco and Nicholas D. Duran with contributions of Rick Dale, Denis O'Hora and Michael J. Spivey.

Maintainer Moreno I. Coco <moreno.cocoi@gmail.com>

Description Extract from two-dimensional x-y coordinates of an arm-reaching trajectory, several dependent measures such as area under the curve, latency to start the movement, x-flips, etc.; which characterize the action-dynamics of the response. Mainly developed to analyze data coming from mouse-tracking experiments.

License GPL (>= 2)

Depends R (>= 3.0.0), pracma

Collate 'angdiffvec.R' 'areaunder.R' 'getmouseDV.R' 'interpltraj.R'
'makerun.R' 'maxdev.R' 'pathoffset.R' 'sampen.R' 'sampenc.R'
'sampense.R' 'trim0.R' 'trim1.R' 'velaj.R' 'xflip.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-29 11:43:37

R topics documented:

mousetrack-package ........................................ 2
getmouseDV ................................................. 3
interpltraj .................................................. 5
mousemove ................................................... 6

Index 8
mousetrack-package  

Mouse-tracking measures from trajectory data.

Description

Extract from two-dimensional (x-y coordinates) of a trajectory several dependent measures, such as area under the curve, latency to start the movement, x-flips, etc. Mainly developed to analyze mouse-tracking data.

Details

Package: mousetrack  
Type: Package  
Version: 1.0  
Date: 2015-01-28  
License: GPL >= 2

getmouseDV: Mother function to extract 40 different measures characterizing the action-dynamics response underlying the arm-reaching trajectory.
interpltraj: Interpolate one (or two-dimensional) trajectory into a fix number of bins.

Author(s)

Moreno I. Coco (moreno.cocoi@gmail.com) and Nicholas D. Duran (nicholas.duran@asu.edu)

References


Examples

data(mousemove)  ## load data
unit = 25; dwellfin = escape = escapeinit = 100  
velajbin = 6

x = mousemove$x; y = mousemove$y;  
counterb = as.character( mousemove$counterb[1] )  
refcounterb = “R”  
t = mousemove$time

ans = getmouseDV(x, y, t, unit, counterb,
**getmouseDV**

`getmousedv(xL yL tL unitL counterbL refcounterbL dwellfinL velajbinL escapeL escapeinit)`

`str(ans)`

---

### Description

Mother function of the package, which computes from 2D (x, y) mouse-tracking trajectories, dependent measures characterizing the underlying action-dynamics patterns.

### Usage

`getmouseDV(x, y, t, unit, counterb, refcounterb, dwellfin, velajbin, escape, escapeinit)`

### Arguments

- **x**: x-coordinate point of the trajectory
- **y**: y-coordinate point of the trajectory
- **t**: a vector with time indexes
- **unit**: the sampling unit, expressed in milliseconds
- **counterb**: the position of the yes-no response button
- **refcounterb**: the position of the yes-no response button, which we want to use as reference to transpose all trajectories on the same side of the display
- **dwellfin**: Region (in pixels) around response button where measures of dwell are computed
- **velajbin**: Number of timesteps used to average velocity
- **escape**: Amount of pixels to escape when trimming the trajectory
- **escapeinit**: Region around origin (in pixels) in which initial angle is measured

### Value

It returns a list with 40 DVs:

- **DVtotaltime**: The total duration of the trajectory
- **DVlatency**: The latency of the start of the movement
- **DVinmot**: The total time of motion
- **DVdwell**: The dwell time to commit to a final response
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVdist</td>
<td>Euclidean distance of the trajectory</td>
</tr>
<tr>
<td>DVdistinmot</td>
<td>Distance traveled outside of the escape region</td>
</tr>
<tr>
<td>DVvelmax</td>
<td>The maximum velocity reached</td>
</tr>
<tr>
<td>DVvelmaxstart</td>
<td>The latency when maximum velocity was observed</td>
</tr>
<tr>
<td>DVaccmax</td>
<td>Maximum acceleration</td>
</tr>
<tr>
<td>DVaccmaxstart</td>
<td>The latency when maximum acceleration was observed</td>
</tr>
<tr>
<td>arclengthtotal</td>
<td>The length of the arc subtending the trajectory after motion was initiated</td>
</tr>
<tr>
<td>maxpathoff</td>
<td>Maximum offset of the trajectory</td>
</tr>
<tr>
<td>DVxflplat</td>
<td>Change in x-direction after escape region: latency</td>
</tr>
<tr>
<td>DVxflpmot</td>
<td>Change in x-direction after escape region: in motion</td>
</tr>
<tr>
<td>DVal1lat</td>
<td>Change of angle-flipping in the dwell region: latency</td>
</tr>
<tr>
<td>DVal1pmot</td>
<td>Change of angle-flipping in the dwell region: in motion</td>
</tr>
<tr>
<td>DVxflpdwl</td>
<td>Change in x-direction in dwell region</td>
</tr>
<tr>
<td>DVflpdwl</td>
<td>Change of angle-flipping in dwell region</td>
</tr>
<tr>
<td>DVxflplat</td>
<td>Change in x-direction after escape region: latency</td>
</tr>
<tr>
<td>DVxflpmot</td>
<td>Change in x-direction after escape region: in motion</td>
</tr>
<tr>
<td>DVxse</td>
<td>Entropy (standard error) along x-axis (default 5 time-points)</td>
</tr>
<tr>
<td>DVyse</td>
<td>Entropy (standard error) along y-axis (default 5 time-points)</td>
</tr>
<tr>
<td>DVae</td>
<td>Entropy along angle-trajectory (default 5 time-points)</td>
</tr>
<tr>
<td>DVase</td>
<td>Entropy (standard error) along angle-trajectory (default 5 time-points)</td>
</tr>
<tr>
<td>trajang</td>
<td>Angles tangent to the trajectory</td>
</tr>
<tr>
<td>DVAUC</td>
<td>Area under the Curve</td>
</tr>
<tr>
<td>DVmaxpull</td>
<td>Maximum pull towards the incorrect response button</td>
</tr>
<tr>
<td>DVmaxpullstart</td>
<td>Latency of pull towards incorrect response button</td>
</tr>
<tr>
<td>DVmaxang</td>
<td>Maximum severity of angle towards incorrect response button while in motion</td>
</tr>
<tr>
<td>DVmaxangstart</td>
<td>Time-point where maximum severity of angle was observed.</td>
</tr>
<tr>
<td>DVinitang</td>
<td>The angle of trajectory after escaping region in absolute value</td>
</tr>
<tr>
<td>DVmaxx</td>
<td>Maximum x-value observed</td>
</tr>
<tr>
<td>DVminx</td>
<td>Minimum x-value observed</td>
</tr>
<tr>
<td>DVmaxy</td>
<td>Maximum y-value observed</td>
</tr>
<tr>
<td>DVminy</td>
<td>Minimum y-value observed</td>
</tr>
<tr>
<td>QLnegmove</td>
<td>Percentage of trajectory not moving towards incorrect response, i.e., negative y values</td>
</tr>
<tr>
<td>QL1</td>
<td>Binary value indicating whether motion time was longer than latency</td>
</tr>
<tr>
<td>QL2</td>
<td>Binary value indicating whether maximum velocity was inside the latency region</td>
</tr>
<tr>
<td>QL3</td>
<td>Binary value indicating whether maximum acceleration is inside the latency region</td>
</tr>
<tr>
<td>QL4</td>
<td>Binary value indicating whether trajectory travels in negative y-values after escaping the latency region</td>
</tr>
</tbody>
</table>
**Note**
A substantial amount of this code has been based on original MATLAB code written by Rick Dale (rdale@ucmerced.edu) and Michael J. Spivey (spivey@ucmerced.edu)

**Author(s)**
Moreno I. Coco (moreno.cocoi@gmail.com) and Nicholas D. Duran (nicholas.duran@asu.edu)

**References**

**Examples**

```r
data(mousemove)
unit = 25; dwellfin = escape = escapeinit = 100
velajbin = 6

x = mousemove$x; y = mousemove$y;
counterb = as.character(mousemove$counterb[1])
refcounterb = "R"
t = mousemove$time

ans = getmousedv(x, y, t, unit, counterb,
                  refcounterb, dwellfin, velajbin,
                  escape, escapeinit)

str(ans)
```

---

**interpltraj**

*Get Mouse Dependent Variables*

**Description**
Interpolate a one-dimensional (angle), or two-dimension (x-y) trajectories to a user specified number of time bins.

**Usage**

`interpltraj(x, y, singlepoint, tmax)`
Arguments

- **x**: x-coordinate point of the trajectory
- **y**: y-coordinate point of the trajectory
- **singlepoint**: a logical flag to indicate whether interpolation is done on a single coordinate point (TRUE) or two points (FALSE)
- **tmax**: the new length of the interpolated trajectory

Value

It returns the interpolated trajectory, either one-dimensional (singlepoint == TRUE), or two-dimensional (singlepoint == FALSE)

Author(s)

Moreno I. Coco (moreno.cocoi@gmail.com)

References


Examples

```r
data(mousemove)
x = mousemove$x; y = mousemove$y;
singlepoint = FALSE; tmax = 101
ans = interp1traj(x, y, singlepoint, tmax)
str(ans)
```

Description

A single trial for a mouse-movement trajectory (x, y, time), and the type of counterbalancing used ("Left") in this specific case.

Usage

`mousemove`
Format

A dataframe with 4 columns (x, y, time, counterbl), and 61 rows.

References

Index

*Topic datasets
  mousemove, 6

*Topic package
  mousetrack-package, 2

*Topic ts, utilities
  mousetrack-package, 2

*Topic ts
  getmouseDV, 3
  interpltraj, 5

getmouseDV, 3

interpltraj, 5

mousemove, 6
mousetrack-package, 2