Package ‘ViSiElse’

October 12, 2022

Type Package

Version 1.2.2

Title A Visual Tool for Behavior Analysis over Time

Description A graphical R package designed to visualize behavioral observations over time. Based on raw time data extracted from video recorded sessions of experimental observations, ViSiElse grants a global overview of a process by combining the visualization of multiple actions timestamps for all participants in a single graph. Individuals and/or group behavior can easily be assessed. Supplementary features allow users to further inspect their data by adding summary statistics (mean, standard deviation, quantile or statistical test) and/or time constraints to assess the accuracy of the realized actions.

URL https://github.com/Re2SimLab/ViSiElse

Author Nastasia Fouret [aut, cph],
Mederic Descoins [aut, cph],
Elodie Garnier [aut, cre, cph],
CEPOI - EA 7388 [cph]

Maintainer Elodie Garnier <e.garnier30@gmail.com>

License AGPL-3

Depends R (>= 3.2.0)

Imports methods (>= 3.2.0), grid (>= 3.2.0), chron (>= 2.3-46), Matrix
(>= 1.2-0), colorspace (>= 1.2-6), stringr (>= 1.0.0), ggplot2,
reshape2

Suggests knitr, rmarkdown

LazyData true

RoxygenNote 6.1.0

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-24 20:00:11 UTC
Description

This dataset shows the actions required to make coffee. The simulated dataset of 10 subjects correspond to the timestamps (in s) of each action. Each value is the time elapse between the beginning of the coffee process and the execution of an action.

Usage

data(coffee)

Format

A data frame with 10 rows and 6 variables:

id Midwife students ID.
coffee Time (in s) when the subject takes the coffee capsule.
fill_coffee Time (in s) when the subject puts the coffee capsule into the machine.
fill_water Time (in s) when the subject fills the water tank of the coffee machine.
push_B Time (in s) when the subject pushes the button to start the machine.
drink Time (in s) when the subject drinks the coffee.

Index

coffee Simulated time data of the coffee process
Examples

```r
data(coffee)
head(coffee)
```

---

**Description**

The method `ConvertFromViSibook` converts a ViSibook in a data.frame object.

**Usage**

```r
ConvertFromViSibook(x)
```

### S4 method for signature 'ViSibook'

```r
ConvertFromViSibook(x)
```

**Arguments**

- `x` a ViSibook object.

**Value**

A data.frame.

**See Also**

- `ViSibook` and see `plot-ViSigrid-method` for examples.

---

**Description**

`ConvertoViSibook` convert a data.frame into a ViSibook object.

**Usage**

```r
ConvertoViSibook(x)
```

**Arguments**

- `x` a dataframe. `x` should contains at least the columns `vars`, `label`, `typeA`, `showorder`, `deb`, `fin`. Optionally, other characteristics can be filled: `GZDebn`, `GZFin`, `Rep- etition`, `BZBeforeDeb`, `BZBeforeFin`, `BZAfterDeb`, `BZAfterFin`, `BZLong` , `BZLtype` .
Value

a ViSibook object.

See Also

See visielse for examples.

Description

Method Dim for ViSibook object.

Usage

```r
## S4 method for signature 'ViSibook'
dim(x)
```

Arguments

x a ViSibook object.

Value

Vector

- [1] The number of actions defined in x.
- [2] The number of characteristics defined in x, its minimum value is 6 and its maximum is 15.

See Also

ViSibook
Description

Method initialize for class ViSibook object.

Usage

```r
## S4 method for signature 'ViSibook'
initialize(.Object, vars, label, typeA, showorder,
          deb, fin, GZDeb, GZFin, Repetition, BZBeforeDeb, BZBeforeFin, BZAfterDeb,
          BZAfterFin, BZLong, BZLtype, NAMES)
```

Arguments

- `vars`: Vector storing names of actions.
- `label`: Vector storing brief description of actions.
- `typeA`: Vector storing type of actions, "l" for long actions, "p" for punctual actions.
- `showorder`: Vector storing order in which actions will be plotted. showorder should be "NA" for actions not to be plotted.
- `deb`: Vector storing, for long actions, the punctual action names that corresponds to its start.
- `fin`: Vector storing, for long actions, the punctual action names that corresponds to its end.
- `GZDeb`: Vector storing punctual actions green zone starting time.
- `GZFin`: Vector storing punctual action green zone ending time.
- `Repetition`: Vector storing if the green zones should be repeated the time interval of repetition.
- `BZBeforeDeb`: Vector storing punctual black zone 1 starting time.
- `BZBeforeFin`: Vector storing punctual black zone 1 ending time.
- `BZAfterDeb`: Vector storing punctual black zone 2 starting time.
- `BZAfterFin`: Vector storing punctual black zone 2 ending time.
- `BZLong`: Vector storing the long action black zone time.
- `BZLtype`: Vector storing the type of the black zone, "time" if the action should be finish at a deadline, "span" if the action should not last more than a period.
- `NAMES`: Vector storing names of slots that are to be considered for `plot-ViSigrid-method`.

Value

a ViSibook object
See Also

See `plot-ViSigrid-method` for examples.

---

**intubation**

*Intubation time data from a simulation of a neonatal resuscitation*

**Description**

Time data from a high-fidelity simulation experiment of a neonatal resuscitation with 37 midwife students. The simulation was video recorded and actions required in the intubation process were tagged. This dataset is the execution time (in seconds) of each action performed by the students.

**Usage**

```r
data(intubation)
```

**Format**

A data frame with 37 rows and 7 variables:

- **id** Midwife students ID.
- **deci_intub** Time (in s) when the student decides to intubate the newborn.
- **stop_ventil** Time (in s) when the student stops the mask ventilation of the newborn.
- **blade_in** Time (in s) when the student inserts the laryngoscope blade in the newborn mouth.
- **insert_tube** Time (in s) when the student inserts the endotracheal tube.
- **blade_out** Time (in s) when the student removes the laryngoscope blade out of the newborn mouth.
- **restart_ventil** Time (in s) when the student restarts to ventilate the newborn through the tube.

**References**


**Examples**

```r
data(intubation)
head(intubation)
```
Description

Method plot for ViSibook object.

Usage

```r
## S4 method for signature 'ViSibook'
plot(x, ncharmax = 10, ncharmaxdelay = 50)
```

Arguments

- **x**: a ViSibook object.
- **ncharmax**: is the maximum number of plotted character for the labels of punctual actions, set to 10.
- **ncharmaxdelay**: is the maximum number of plotted character for the labels of long actions, set to 50.

See Also

- ViSibook, visielse

Method plot for ViSigrid object. This method provides a graphic of raw data during experimental observations of the realization of a procedure like a medical algorithm. It graphically presents an overview of individuals and group actions usually acquired from timestamps during video recorded sessions.

Usage

```r
## S4 method for signature 'ViSigrid'
plot(x, scal.unit.tps = 10, unit.tps = "s",
     main = "", ncharlabel = 30, size.main = 12, Fontsize.title = 11,
     Fontsize.label.Action = 11, Fontsize.label.Time = 11,
     Fontsize.label.color = 9, col.main = "black", col.grid = "grey",
     colgreenzone = "green", colblackzone = "black", alphainf = 0.8,
     alphasup = 1, alphaZones = 0.2, vp0h = 0.6, vp0w = 0.6,
     lina = 0.7, rcircle = 15, lwdline = 2, lwd.grid = 1,
     lty.grid = 1)
```
Arguments

x A ViSigrid object built using the `visielse` function.
scal.unit.tps Unity of time for the grey grid legend.
unit.tps Unit of time (s,min,...).
main Title.
ncharlabel Maximum number of plotted characters for labels of actions.
size.main Title size.
Fontsize.title Fontsize of the title.
Fontsize.label.Action Fontsize of labels of plotted actions.
Fontsize.label.Time Fontsize of the time axis.
Fontsize.label.color Fontsize of legends.
col.main Title color.
col.grid Color of the legend box.
colgreenzone Color of the green zones.
colblackzone Color of black zones.
alphainf Alpha of informers circles.
alphasup Alpha of supplementary times.
alphaZones Alpha of green and black zones.
vp0h Height of the main plot window, <1.
vp0w Width of the main plot window, <1.
linA Height of the plotting area in each actions lines, <1.
rcircle circle radius of informers circles.
lwdline line width of lines linking the 3 informers circles.
lwd.grid Lines width of the legend box.
lty.grid Lines type of the legend box.

See Also

`ViSigrid, ViSibook, visielse`. 
**Method print for ViSibook object.**

**Usage**

```r
## S4 method for signature 'ViSibook'
print(x)
```

**Arguments**

- `x`: a ViSibook object.

**See Also**

ViSibook, visielse, and see plot-ViSigrid-method for examples.

**Method set for ViSibook object.**

**Description**

Method set for ViSibook object.

**Usage**

```r
## S4 replacement method for signature 'ViSibook,numeric,numeric,ANY'
x[i, j] <- value

## S4 replacement method for signature 'ViSibook,missing,numeric,ANY'
x[i, j] <- value

## S4 replacement method for signature 'ViSibook,numeric,missing,ANY'
x[i, j] <- value
```

**Arguments**

- `x`: a ViSibook object.
- `i` and `j`: a numeric.
- `value`: object to allocate.
shoppingBehavior

Value

a ViSibook object.

See Also

ViSibook

shoppingBehavior Simulated online shopping behavior time data

Description

This dataset shows the buying process of consumers over internet based on a 5-steps model: need recognition, information search, evaluation, purchase decision, and post-purchase behavior. This simulated dataset of 100 subjects correspond to the timestamps (in s) of each action of the model (except for the post-purchase behavior) executed by the subjects.

Usage

data(shoppingBehavior)

Format

A data frame with 100 rows and 7 variables:

- id  Customer ID.
- need Time (in s) when the customer decides he/she needs an item.
- start_search Time (in s) when the customer starts to search for the item.
- stop_search Time (in s) when the customer stops to search for the item.
- start_eval Time (in s) when the customer starts to evaluate the item.
- stop_eval Time (in s) when the customer stops to evaluate the item.
- deci Time (in s) when the customer decides to buy the item.

References


Examples

data(shoppingBehavior)
head(shoppingBehavior)
**Method show-ViSibook**

**Description**
Method show for ViSibook object.

**Usage**
```r
## S4 method for signature 'ViSibook'
show(object)
```

**Arguments**
- `object` a ViSibook.

**See Also**
- `ViSibook`.

**Method show-ViSigrid**

**Description**
Method show for ViSigrid object.

**Usage**
```r
## S4 method for signature 'ViSigrid'
show(object)
```

**Arguments**
- `object` a ViSigrid.

**See Also**
- `ViSigrid` and see `plot-ViSigrid-method` for examples.
Method summary-ViSigrid

Description
Method summary for ViSigrid object.

Usage
```r
## S4 method for signature 'ViSigrid'
summary(object)
```

Arguments
- `object`: a ViSigrid.

Value
- list
  - `punctual`: summary of punctual actions (typeA=="p").
  - `longs`: summary of long actions (typeA=="p").

See Also
- `ViSigrid`, `visielse,ViSibook`. and see `plot-ViSigrid-method` for examples.

Simulated time data of the actions performed in a typical day

description

This dataset shows the actions usually performed during a typical day. The simulated dataset of 100 subjects correspond to the timestamps (in min) of each action of the day, from midnight to midnight. Each value is the time elapsed between the beginning of the day (midnight) and the execution of the action.

Usage
```r
data(typDay)
```
Format

A data frame with 100 rows and 15 variables:

- **id**: Midwife students ID.
- **start_sleep**: Time (in min) when the subject is sleeping. All subjects are set to 0 (0:00 or midnight) as the dataset shows a day from midnight to midnight.
- **stop_sleep**: Time (in min) when the subject stops to sleep.
- **wake_up**: Time (in min) when the subject wakes up (same values as stop_sleep).
- **shower**: Time (in min) when the subject takes a shower.
- **breakfast**: Time (in min) when the subject eats breakfast.
- **start_work**: Time (in min) when the subject starts working.
- **start_lunch**: Time (in min) when the subject starts to eat lunch.
- **stop_lunch**: Time (in min) when the subject finishes his lunch.
- **stop_work**: Time (in min) when the subject stops to work.
- **pickup_kids**: Time (in min) when the subject picks up his kids.
- **start_cook**: Time (in min) when the subject starts cooking.
- **stop_cook**: Time (in min) when the subject stops cooking.
- **go_sleep**: Time (in min) when the subject goes to sleep.
- **first_coffee**: Time (in min) when the subject drinks his first coffee of the day.

References


Examples

```r
data(typDay)
head(typDay)
```

Description

Class **ViSibook** defines the structure of the process to be plotted.
Slots

dvars a vector storing names of actions.

label a vector storing brief description of actions.

typeA a vector storing type of actions, "l" for long actions (which have a starting time and an ending time), "p" for punctual actions.

showorder a vector storing order in which actions will be plotted. When an action should not be plotted, showorder should be NA.

deb a vector.
  • Long actions deb stores the punctual action names that delimit the long actions beginning.
  • Punctual actions NA.

fin a vector.
  • Long actions fin stores the punctual action names that delimit the long actions ending.
  • Punctual actions NA.

GZDeb a vector, optional, GZdeb stores punctual actions green zone starting time.

GZFin a vector, optional, GZFin stores punctual actions green zone ending time.

Repetition optional a vector, optional, When a green zone is defined, Repetition stores the duration of the time interval between green zones.

BZBeforeDeb a vector, optional, BZBeforeDeb a vector storing punctual black zone 1 starting time.

BZBeforeFin a vector, optional, BZBeforeFin storing punctual black zone 1 ending time.

BZAfterDeb a vector, optional, BZAfterDeb stores punctual black zone 2 starting time.

BZAfterFin a vector, optional, BZAfterFin stores punctual black zone 2 ending time.

BZLong a vector, optional, BZLong stores the long action black zone time.

BZLtype a vector, optional, BZLtype stores the type of the black zone, "time" if the action should be finish at a time, "span" if the action should be finish in a time.

NAMES a vector storing names of slots that are to be defined.

See Also

visielse for examples.

ViSibookfromDATA Function

Description

ViSibookfromDATA build an object class ViSibook from observational data. The process is the ordered list of punctual actions given by the columns names of X.

Usage

ViSibookfromDATA(X, idsubject = 1)
Arguments

X  data.frame.

idsubject  numeric indicates the number of the column of X which stores id.

Value

a ViSibook corresponding to the dataset X.

Description

visielse plots the graphic from time data and build an object class ViSigrid with, at least, the time data of each punctual action defined in the ViSibook for all subjects.

Usage

visielse(X, book = NULL, is.ViSibook = FALSE, doplot = TRUE, Xsup = NULL, method = "global", group = NULL, grwithin = NULL, informer = "median", tests = TRUE, threshold.test = 0.01, quantity = "N", pixel = 20, t_0 = 0, sorted.line = TRUE, decrgr2 = FALSE, max_tps = NULL, colvect = NULL, ncolvect = NULL, times = FALSE, timeformat = c("hh:mm:ss"), idsubject = 1)

Arguments

X  A data.frame or matrix. X stores punctual action timestamps. The actions are defined in book, and X columns names should correspond to the slot "vars" of book. X must also have a column to identify individuals (id column).

book  A data.frame or a ViSibook or NULL. book stores the process structure.

• If it is a data.frame it should contains at least the columns vars, label, typeA, showorder, deb, fin . Optionally, other characteristics can be filled : GZDebn, GZFin, Repetition, BZBeforeDeb, BZBeforeFin, BZAfter-Deb, BZAAfterFin, BZLong , BZLtype .

• If it is a ViSibook it should correspond to the columns names of X.

• If it is NULL the process is the ordered list of punctual actions given by the columns names of X.

is.ViSibook  A logical

• FALSE if book is a data.frame or NULL.

• TRUE is book is a ViSibook.

doplot  A logical If FALSE, the graphic is not plotted.

Xsup  A data.frame or matrix storing supplementary time data, all individuals in Xsup must be in X.
method  In \{ "global", "cut", "join", "within" \}. method specifies the plotting method, see details. If group is NULL, method is set to "global".

group  A factor with two levels. group indicates the group attributed to the individuals, it has same the length as the number of rows of \(X\).

grwithin  A level of group. If method is set to within, grwithin specifies the group to consider.

informer  In \{ "NULL", "median", "mean" \}. If informer is set to "median" the median and quartiles are computed, if it is set to "mean" the mean and standard deviation are. If informer is NULL no indicators are computed.

tests  A boolean. When informer is not NULL and group is defined, if tests is TRUE, tests are computed to compare groups. If the parameter informer is set to "mean", the function wilcox.test() is used, if informer is set to "median" the function mood.test() is used.

threshold.test  A numeric between 0 and 1. threshold.test is the value of the p-value under which the H0 hypothesis of the test is rejected when tests is TRUE.

quantity  In \{ "N", "dens" \}. quantity allows choosing the quantity represented for punctual action When quantity is set to "N" the number of individuals is considered. Otherwise when it is set to "dens" proportion of individuals is considered instead. If group is defined and method set to "cut" or "within", this proportion is calculated regarding each represented group.

pixel  An integer. It is the number of unit of time under which individuals are aggregated in the plot.

t_0  either 0, either a value of the slot "vars" in book, t_0 indicates the starting time to plot.

sorted.line  A boolean. When sorted.line is TRUE, it allows long actions to be sorted by starting time.

decrgr2  A boolean. When sorted.line is TRUE and decrgr2 is TRUE, long actions of the second group are plotted in decreasing order by starting times.

max_tps  A numeric, \(>0\). max_tps is the maximum time used to build the grid in the plot. max_tps is useful when Xsup is given. If max_tps is NULL it is automatically computed.

colvect  A matrix containing colors. Colors are automatically computed if colvect is NULL. If group is not NULL colvect should have two rows otherwise one.

ncolvect  A numeric. ncolvect indicates the number of columns of colvect. Its default setting is \(\text{dim}(X)[1]\). ncolvect is considered only if colvect is NULL.

times  A boolean. If times is TRUE, it indicates that \(X\) contains data in a time format.

timeformat  time format. If times is TRUE.

idsubject  An integer between 1 and \(\text{dim}(X)[2]\). idsubject indicates the number of the column of \(X\) that contains individuals id numbers.

Details

* method
- **global**: The plot of the ViSigrid object will not consider the parameter group and plot indistinctly all individuals.
- **cut**: In the plot of the ViSigrid object, each group will be plotted separately, one under the other with different colors.
- **join**: In the plot of the ViSigrid object, groups are spatially mixed but they are represented by different colors.
- **within**: In the plot of the ViSigrid object, all individuals are plotted together then the group specified in `grwithin` is plotted another time underneath.

• **informer**
  The parameter `informer` allows users to choose the statistics to be plotted. `informer` can take three values:
  - **median**: Median and quartiles are calculated for each action, using the function `quantile` from the package stats. This is the default value.
  - **mean**: Mean and standard deviation are calculated for each action, using the functions `mean` and `var` from the package stats.
  - **NULL**: no indicators are computed.
When a group is defined, statistics are calculated per group if the method cut or within is chosen.
When plotting the `ViSigrid` object, statistics for punctual actions are represented by white circles linked by a line. For long action, only a black line is plotted starting at the median (or mean) value of the punctual action staring times. The line length represents the median (or mean) of the long action duration. Informers are computed directly on the given matrix for punctual action. And for long actions, it is based on the difference between the punctual action defining its beginning and the one defining its ending.

• **tests and threshold.test**
  As for the parameter `informer`, tests are computed on the given matrix or data.frame `X` for a punctual action. And for a long action, it is calculated on its difference between its beginning and its ending punctual actions. In `plot-ViSigrid-method`, results of the tests are represented by a star only when the resulted p-value is below or equal to value defined by the parameter `threshold.test`.

• **pixel**
  The parameter `pixel` represents the number of unit of time under which individuals are aggregated for punctual action in the plot. When the parameter `pixel` is too small the information represented will be too much aggregated to allow interpretation.
  For punctual actions data are aggregated in a matrix $M$. The number of row of $M$ is the number of action and its number of columns is $[(\text{max}(X) - t_0)/\text{pixel}]$.
  $M_{i,j}$ contains the number of observations of the $i$-th punctual action (by the order of the ViSibook object) between $t_0 + (j - 1)\text{pixel}$ included and $t_0 + j \ast \text{pixel}$ excluded.

• **t_0**
  The origin of the graphic can be set using the parameter `t_0`. There is two ways to define it:
  - A number: set to 0. It can be change at convenience, but for long actions black zones will not be drawn, and for punctual actions black and green zones will not be translated.
  - The name of a punctual action: To set the origin of the graphic to the moment when the action was done for each individual. Black and green zones will not be translated as well.
x can also has the columns: `GZDebn`, `GZFin`, `Repetition`, `BZBeforeDeb`, `BZBeforeFin`, `BZAfterDeb`, `BZAfterFin`, `BZLong`, `BZLtype`
Value

a ViSigrid object.

See Also

Classes ViSigrid and ViSibook. The method plot for ViSigrid object plot-ViSigrid-method for examples.

Examples

coffee <- c( 58, 11, 5, 53, 53, 59, 24, 59, 46, 20)
fill_coffee <- c(162, 57,103,154,165,132, 74, 107, 104, 93)
fill_water <- c( 66, 92,54, 78, 74, 114, 91, 129, 71, 56)
push_B <- c( 74, 99, 62, 84, 83, 120, 95, 129, 80, 63)
drink <- c( 472, 176, 475, 283, 265, 207, 234, 184, 490, 520)
X <- data.frame(id = seq(1,10), coffee, fill_coffee,fill_water,push_B,drink)
library(ViSiElse)
visi1 <- visielse(X)

#### Changing the pixel of time
visi1 <- visielse(X, pixel = 10)

# Plot the mean and standart deviation
visi1 <- visielse(X,informer = "mean")

#### Do not plot indicators
visi1 <- visielse(X,informer = NULL)

# Extraction of the visibook from the data
visi1 <- visielse(X,informer = NULL, doplot = FALSE)
book <- visi1@book
plot(book)

#### Changing labels
book[,2]<- c("Taking the coffee",
  "Fill the machine with coffee",
  "Fill the tank with water",
  "Push the Button",
  "Drink the coffee")
plot(book)
visi1 <- visielse(X, book=book, is.ViSibook = TRUE,informer = NULL)

#### Change the order of Actions in the process
book[,4]<- c(5,1,2,4,3)
plot(book)
visi1 <- visielse(X, book=book, is.ViSibook = TRUE)
### Adding a long Actions

```r
visi1 <- visielse(X)
book <- ConvertFromViSibook(visi1@book) # Convert book into data.frame
add_delay <- c("delay_coffee_push","Preparation","l","6","coffee","push_B")
book[6,] <- add_delay
book
```

### ViSiElse representation of long actions

```r
visi2 <- visielse(X=X, book=book, informer=NULL)
```

#### Green & Black zones

```r
book$GZDeb <- c(NA,60,NA,NA,NA,NA)
book$GZFin <- c(NA,120,NA,NA,NA,NA)
book$BZBeforeDeb <- c(NA,0,NA,NA,NA,NA)
book$BZBeforeFin <- c(NA,30,NA,NA,NA,NA)
book$BZAfterDeb <- c(NA,180,NA,NA,NA,NA)
book$BZAfterFin <- c(NA,Inf,NA,NA,NA,NA)
book$BZLong <- c(rep(NA,5),150)
book$BZLtype <- c(rep(NA,5),"time")
visi1 <- visielse(X, book=book, informer = NULL)

book$BZLtype <- c(rep(NA,5),"span")
visi1 <- visielse(X, book=book, informer = NULL)
```

#### Group

### Method : Cut

```r
group <- c("group2","group1","group2","group1","group1",
   "group2","group1","group1","group1","group2")
visi1 <- visielse(X,group=group, book=book, informer = NULL, method = "cut")
```

```
visi1 <- visielse(X,group=group, book=book, informer = NULL, method = "join")
visi1 <- visielse(X,group=group, book=book, informer = NULL, method = "within",grwithin = "group1")
```

---

**Description**

Class ViSigrid defines the structure of the process to be plotted.
Slots

- `MATp` A "dgCMatrix". It stores the grid for all punctual actions in the book.
- `MATpsup` A "dgCMatrix". It stores the grid for all punctual actions in the book corresponding to the supplementary times.
- `idsup` A "vector". It stores individuals id having supplementary times.
- `colvect` A "matrix". Matrix with colors to use.
- `L` A "data.frame". It stores the data corresponding to long actions having a showorder.
- `idsort` A "matrix". For all long actions, it stores the order of individuals in which each actions will be plot.
- `BZL` A "dgCMatrix". It stores black zones for long actions, calculated for each individuals.
- `Lsup` A "data.frame" for the long actions having a showorder and supplementary times defined, it stores the data corresponding to those actions.
- `book` A "ViSibook". It stores the structure of the grid for the plot.
- `group` A "factor". It stores the group for each individuals.
- `vect_tps` A "vector". It stores the times vector mapping the grid.
- `informers` A "matrix". It stores the statistics (mean, median or NULL) by actions.
- `testsP` A "vector". Results of tests p.value<threshold.test.
- `parameters` A "list". It stores the parameters put in the `visielse` function.

See Also

`visielse, plot, ViSigrid-method, ViSibook`

[.,ViSibook, numeric, missing, ANY-method

Method get for ViSibook object.

Description

Method get for ViSibook object.

Usage

```r
## S4 method for signature 'ViSibook, numeric, missing, ANY'
x[i, j, drop = TRUE]
```

```r
## S4 method for signature 'ViSibook, missing, numeric, ANY'
x[i, j, drop = TRUE]
```

```r
## S4 method for signature 'ViSibook, numeric, numeric, ANY'
x[i, j, drop = TRUE]
```
Arguments

- x: a ViSibook object.
- i: a numeric.
- j: a numeric.
- drop: = TRUE.

Value

obj.

See Also

ViSibook.
## Index

* **datasets**
  - coffee, 2
  - intubation, 6
  - shoppingBehavior, 10
  - typDay, 12
  - \([,\text{Visibook,missing,numeric,ANY-method}\]
    \(\text{[,ViSibook,numeric,missing,ANY-method]}\), 20
  - \([,\text{ViSibook,missing,numeric-method}\]
    \(\text{[,ViSibook,numeric,missing,ANY-method]}\), 20
  - \([,\text{ViSibook,numeric,missing-method}\]
    \(\text{[,ViSibook,numeric,missing,ANY-method]}\), 20
  - \([,\text{ViSibook,numeric,numeric-method}\]
    \(\text{[,ViSibook,numeric,missing,ANY-method]}\), 20
  - \([,\text{ViSibook,numeric,numeric-method}\]
    \(\text{[,ViSibook,numeric,missing,ANY-method]}\), 20
  - \([-\text{-},\text{ViSibook,missing,numeric,ANY-method}\]
    \(\text{set-ViSibook-method}\), 9
  - \([-\text{-},\text{ViSibook,numeric,missing,ANY-method}\]
    \(\text{set-ViSibook-method}\), 9
  - \([-\text{-},\text{ViSibook,numeric,numeric,ANY-method}\]
    \(\text{set-ViSibook-method}\), 9

- coffee, 2
- ConvertFromViSibook
  - \(\text{ConvertFromViSibook-ViSibook-method}\), 3
- ConvertFromViSibook, ViSibook-method
  - \(\text{ConvertFromViSibook-ViSibook-method}\), 3
- ConvertFromViSibook-ViSibook-method, 3
- ConvertToViSibook, 3
- dim, ViSibook-method

<table>
<thead>
<tr>
<th>dim-ViSibook-method, 4</th>
<th>initialize, ViSibook-method, 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>intubation, 6</td>
<td>intubation, 6</td>
</tr>
<tr>
<td>plot, ViSibook-method, 7</td>
<td>plot-ViSibook-method, 7</td>
</tr>
<tr>
<td>plot-ViSigrid-method, 7</td>
<td>print-ViSibook-method, 9</td>
</tr>
<tr>
<td>print-ViSibook-method, 9</td>
<td>set-ViSibook-method, 9</td>
</tr>
<tr>
<td>shoppingBehavior, 10</td>
<td>show, ViSibook-method, 11</td>
</tr>
<tr>
<td>show, ViSigrid-method, 11</td>
<td>show-ViSigrid-method, 11</td>
</tr>
<tr>
<td>summary, ViSigrid-method, 12</td>
<td>summary-ViSigrid-method, 12</td>
</tr>
<tr>
<td>typDay, 12</td>
<td>ViSibook, 3, 4, 7–12, 18, 20, 21</td>
</tr>
<tr>
<td>ViSibook (ViSibook-class), 13</td>
<td>ViSibookfromDATA, 14</td>
</tr>
<tr>
<td>ViSigrid, 8, 11, 12, 17, 18</td>
<td>visielse, 4, 7–9, 12, 14, 15, 20</td>
</tr>
<tr>
<td>ViSigrid (ViSigrid-class), 19</td>
<td>ViSigrid-class, 19</td>
</tr>
</tbody>
</table>