Package 'surveyeditor'

June 3, 2015

Type Package

Title Generate a Survey that can be Completed by Survey Respondents

Version 1.0

Date 2015-05-14

Author Char Leung

Maintainer Char Leung <charleung@hotmail.fr>

Description Help generate slides for surveys or experiments.
The resulted slides allow the subject to respond with the use of the mouse (usual keyboard input is replaced with clicking on a virtual keyboard on the slide). Subjects' responses are saved to the user-specified location in the form of R-readable text file. To allow flexibility, each function in this package generates a particular type of slides thus general R function writing skills are required to compile these edited slides.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-03 19:03:49

R topics documented:

surveyeditor-package *Generate a Survey that can be Completed by Survey Respondents*

Description

Help generate slides for surveys or experiments. The resulted slides allow the subject to respond with the use of the mouse (usual keyboard input is replaced with clicking on a virtual keyboard on the slide). Subjects' responses are saved to the user- specified location in the form of R-readable text file. To allow flexibility, each function in this package generates a particular type of slides thus general R function writing skills are required to compile these edited slides.

Details

The DESCRIPTION file:

Package:	surveyeditor
Type:	Package
Title:	Generate a Survey that can be Completed by Survey Respondents
Version:	1.0
Date:	2015-05-14
Author:	Char Leung
Maintainer:	Char Leung <charleung@hotmail.fr>
Description:	Help generate slides for surveys or experiments. The resulted slides allow the subject to respond with the use o
License:	GPL (>=2)

Index of help topics:

cover	Cover page
identity	Subject id configuration
read.write	Dynamic survey data storage function
slide	Survey/questionnaire slides
surveyeditor-package	Generate a Survey that can be Completed by Survey Respondents

This package aims to serve as an editor for generating slides for surveys or experiments. The resulted slides allow the subject to respond with the use of the mouse (usual keyboard input is replaced with clicking on a virtual keyboard on the slide). Subjects' responses are saved to the user- specified location in the form of R-readable text file. To allow flexibility, each function in this package generates a particular type of slides thus general R function writing skills are required to compile these edited slides.
Main features include:

- Edit texts in each slide as well as their font size and colour.
- Insert tables and plots (point, line and histogram).

- Allow time-restricted display (only for slides not requiring responses).
- Different types of responding such as numerical input, multiple choices (can have more than one selections), Likert scale type questions, number line drawing type question (eg, used in Kanayet & Opfer 2009 and Siegler & Opfer 2003) and Gabor-Granger method (Gabor & Granger 1966, 1979).
- Automatically save responses to a file to a user-specified location in the form of R-readable text file.

There are in general 4 functions in this package:
`cover` : serves as the front or back cover page of the survey. For example, you may wish to have a greeting slide before the survey begins.
`identity` : serves as gathering the id of the subject. Automatic id generation is also allowed.
`read.write` : serves as saving the responses to a nominated file.
`slide` : generates slides to be displayed to the subjects as well as allowing the subjects to respond to designated questions. This is the main function in this package.

The editor of the survey should write a function (a compiler) to compile different functions into a meaningful survey. A recommended format is shown below,

```
compiler<-function(){
cover
identity
slide
slide
...
read.write
cover}
```

The first `cover` serves as the front page then the identity of the subject is configured in `identity`. `slide` is the main part of the survey that display information and allow the subject to respond. Responses are then saved to the nominated file through the `read.write` function. The survey ends in the second `cover` informing the subject the completion.
More specifically, the below is an example,

```
sample.survey<-function(){
link<-"C:"
repeat {
cover(type="front")
id<-identity(link=link, ...)
Q1<-slide(id=id, ...)
Q2<-slide(id=id, ...)
...
read.write(rbind(Q1,Q2,...),link=link)
cover(type="back")
} }
```

The `repeat` is used to let the next subject take the survey and the response will be incorporated into the user-specified location.

Notes: Here is some recommendations for the best performance.

- **The graphic device should be adjusted to full-screen size or some texts may appear outside the box.**
- **It is a good idea to inform the subjects (preferably by stating in a slide titled "Instructions") that all responses of the previous slides cannot be altered and that clicking the green button in any slide will proceed to the next slide allowing no further alteration to the response. In fact this is the reason only certain buttons are green.**

Author(s)

Char Leung
Maintainer: Char Leung <charleung@hotmail.fr>

References

Gabor, G & Granger, CWJ (1966) "Price as an indicator of Quality: Report on an Enquiry", *Economica*, **33**(129), pp.43-70.
Gabor, G & Granger, CWJ (1979) "The Attitude of the Consumer to Prices", *Management Decision*, **17**(8), pp.619-634.
Kanayet, F & Opfer, J (2009) "Why Children's Number-line Estimates Follow Fechner's Law", *Cognitive Science Conference Proceedings 2009*, pp.1936-41.
Siegler, RS & Opfer, JE (2003) "The Development of Numerical Estimation: Evidence for Multiple Representations of Numerical Quantity", *Psychological Science*, **14**(3), May 2003, pp.237-243.

`cover` *Cover page*

Description

Front and back cover page of the survey.

Usage

```
cover(type = c("front", "back"), content, col, size, loc, time = NULL)
```

Arguments

`type`	To indicate if the cover is a front of a back cover page. If back page, `time` must be set. See Details for other differences.
`content`	Texts to be displayed in the slide. It can be a vector `c( )` containing multiple items.
`col`	The colour of `content` hence its length must be the same as that of `content`.
`size`	The font size of `content` hence its length must be the same as that of `content`.
`loc`	The vertical location of `content`. Must be a number within the interval [0,100]. Its length must be the same as that of `content`.
`time`	The time (in seconds) that the slide is displayed for. Works only `type="back"`. See Details.

Details

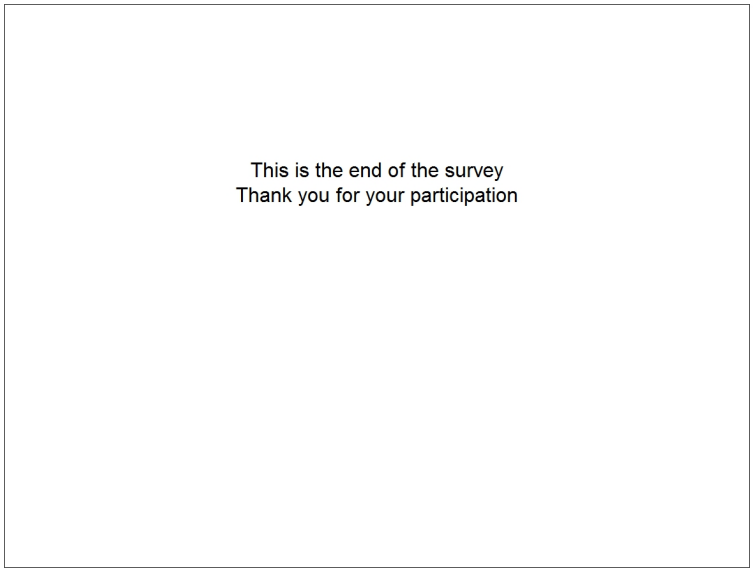
If it is a front cover page (i.e. `type="front"`) a green button is shown in the bottom of the slide to allow proceeding to the next slide. The reason that `time` only works for `type="back"` is that the back cover page closes in a few seconds so the entire survey program starts again for the next subject if the command `repeat{}` is used. For example,

```
repeat{cover(type="front",...)
identity(...)
slide(...)
slide(...)
...
read.write(...)
cover(type="back",time=10,...)
}
```

Value

A slide is displayed on the graphic device and no output in the console.

Returning graphics of Examples



Author(s)

Char Leung

See Also

identity,read.write,slide

Examples

```
## Not run: cover(type="front",content=c("Inflation expectation experiment","Please wait
for further instructions"),col=c("blue","black"),size=c(2.5,1.5),loc=c(60,30))
## End(Not run)
## Not run: cover(type="back",content="This is the end of the survey\nThank you for your
participation",col="black",size=2,loc=70,time=3)
## End(Not run)
```

`identity` *Subject id configuration*

Description

To configure the identity of the subject assuming that there is more than one subject in the survey. This is done by two methods: (1) manual configuration generates a slide to ask the subject for an id. The question asking the subject must be edited and (2) automatic configuration requires the editor to insert a file location where a file is read and the previous id is sought. Then current id is the sought previous id "+1". This means that the resulted id is a set of natural number. This option is best used when the real identity of the subject is not to be disclosed.

Usage

```
identity(type = c("manual", "auto"), ask = NULL, link = NULL)
```

Arguments

type	Either "manual" or "auto" to indicate manual or automatic configuration.
ask	The question asking for the id to be displayed in the slide if type="manual". A string thus must be quoted. The question will be allocated at the top left hand corner. Font size and colour are fixed at 2 and "black".
link	The location of the file that will be sought for the previous id if type="auto". A string thus must be quoted.

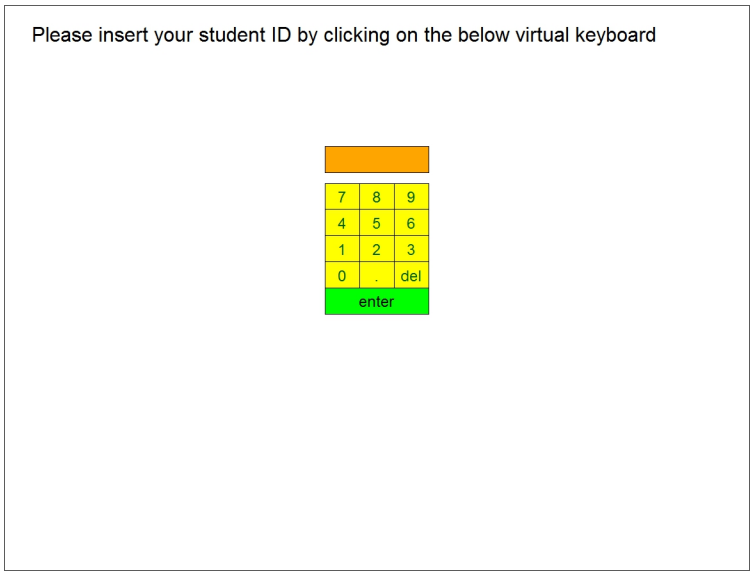
Details

If type="auto", there are three possible the outcomes in seeking the file link.
(1) The file link exists and it has column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response** (this is the universal format of the responses returned by slide). In this case, the previous id is successfully sought and the automatically generated id is the previous id "+1".
(2) The file link does not exist. In this case, the file link is generated with the column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response**. The automatically generated id is made to 1.
(3) The file link exists but this not have column names link is generated with the column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response**. This is usually the case where the file name is already in use. Either the file name of the existing file or link should be altered.

Value

The id is returned on the R console regardless of type. If type="manual", a slide is shown to ask the subject to insert id. ask is displayed at the top left hand corner and a virtual keyboard is placed in centre of the slide for the subject to enter the id.

Returning graphics of examples



Author(s)

Char Leung

See Also

slide

Examples

```
## Not run: identity(type="manual",ask="Please insert your student ID by clicking on the below
virtual keyboard")
## End(Not run)
## Not run: identity(type="auto",link="R:/survey.txt")
```

---

read.write *Dynamic survey data storage function*

Description

This function saves matrix-type responses to a nominated file. If the nominated file already exists and the responses of previous subjects are in the file, current matrix-type bundled responses will be incorporated into this file.
Note: one should bundle multiple responses (i.e. responses from multiple slide) by using the rbind function.

Usage

```
read.write(rbind.result, link)
```

Arguments

rbind.result	Bundled responses formed by using rbind to bind the responses from slide. As seen in slide, slide returns a 5-column matrix with column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response**. Therefore, responses from multiple slide must be compiled with the use of rbind.
link	The location of the file that rbind.result will be incorporated into.

Details

There are three possible the outcomes in seeking the file, link.
(1) The file link exists and it has column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response** (the universal format of the responses returned by slide). In this case, rbind.result is incorporated into link.
(2) The file link does not exist. In this case, the file link is generated with the column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response** and rbind.result is incorporated into link.
(3) The file link exists but this not have column names **ID**, **Question.number**, **type**, **Condition.Likert** and **Response**. This is usually the case where the file name in link is already in use. Either the file name of the existing file in the directory link or the link in this function should be altered to avoid confusion.

Value

Nothing is shown in the R console or the graphic device. Yet the file link will be altered (as rbind.result is incorporated) or will be created.

Author(s)

Char Leung

See Also

slide

Examples

```
## Not run: link<-"R:/test.txt"
## Not run: id<-identity("auto",link=link)
## Not run: Q1<-slide(id=id,type=7,title="Question 1",content="Would you take the milk if
it costs $",h=0,v=80,size=2,col="black",initial=3,increment=1,order="descending"
,box.level=70)
## End(Not run)
## Not run: Q2<-slide(id=id,type=5,title="Question 2",content=c("The market price is $"
,"(0 refers to 'Strongly disagree' ... 5 refers to 'Strongly agree')"),h=c(0,0)
,v=c(80,70),size=c(2,1.5),col=c("black","orange"),cond=seq(10,12,0.5),scale=seq(0,5,0.5)
,box.level=50)
```

```
## End(Not run)
## Not run: read.write(rbind(Q1,Q2),link=link) #rbind is used to form a bundled
matrix-type responses
## End(Not run)
```

---

slide *Survey/questionnaire slides*

Description

This function generates slides for displaying information (including questions to be asked) and for the subjects to insert answers. 7 types of slides are available (the argument `type` specifies this; see Arguments below).

For `type=1` or 2, the slide is solely for display purpose (e.g. showing the instructions of the survey or background information) and responses from the subject are not needed.

For other values of `type`, a 5-column matrix is returned. The column names are **ID**, **Question.number**, **type**, **Condition.Likert** and **Response**. Also see Values below. The number of rows is determined by the number of responses gathered by the slide. For example, for `type=2` only one reponse is allowed to be given by the subject therefore there will be a 1x5 matrix returned.

The below table (in a few partitions) shows the number of allowed responses for different values of `type`,

type=	Slide type	Allowed responses
1	Time-restricted display	0
2	Time-unrestricted display	0
3	Numerical response display	1
4	Multiple choice response display	Freely set and up to the number of available choices (`n.select`)
5	Sequential Likert-scale response display	The number of conditions given (the length of `cond`)
6	"Number line marking" response display	Freely set (`n.select`)
7	Gabor-Granger method display	1

Usage

```
slide(id, type, title, content, h, v, size, col, time = NULL, interval = c(-Inf, Inf),
isinteger = F, choices = NULL, layout = NULL, n.select = NULL, box.level = NULL,
cond = NULL, scale = NULL, grid = NULL, range = NULL, unit = NULL, initial = NULL,
increment = NULL, order = c("ascending", "descending"), Extra = NULL, Line.draw = NULL)
```

Arguments

Which arguments are required depend on the slide type (the value of `type`). The below table gives a summary as well as the input type of each argument.

type=	id	title	content	h	v	size
1	Required	Required	Optional	Optional	Optional	Optional
2	Required	Required	Optional	Optional	Optional	Optional
3	Required	Required	Optional	Optional	Optional	Optional
4	Required	Required	Optional	Optional	Optional	Optional
5	Required	Required	Optional	Optional	Optional	Optional
6	Required	Required	Optional	Optional	Optional	Optional
7	Required	Required	Optional	Optional	Optional	Optional
Input type:	numeric(1)	character(1)	character()	numeric()	numeric()	numeric()

type=	col	time	interval	isinteger	choices
1	Optional	Required			
2	Optional				
3	Optional		Optional	Optional	
4	Optional				Required
5	Optional				
6	Optional				
7	Optional				
Input type:	character()	numeric(1)	numeric(2)	logical(1)	character() or numeric( )

type=	layout	n.select	box.level	cond
1				
2				
3			Required	
4	Required	Required	Required	
5			Required	Required
6		Required	Required	
7			Required	
Input type:	numeric(2)	numeric(1)	numeric(1) or numeric(2) if `type=3`	numeric() or character()

type=	scale	grid	range	unit
1				
2				
3				
4				
5	Required			
6		Required	Required	Optional
7				
Input type:	numeric() or character()	numeric(1)	numeric(3)	character(1) or numeric(1)

type=	initial	increment	order
1			
2			
3			
4			
5			
6			
7	Required	Required	Required
Input type:	numeric(1)	numeric(1)	"ascending" or "descending"

type=	Extra	Line.draw
1	Optional	Optional
2	Optional	Optional
3	Optional	Optional
4	Optional	Optional
5	Optional	Optional
6	Optional	Optional
7	Optional	Optional
Input type:	list(list(),list(),...)	character() commands to be executed

numeric(1); The id of the subject. It is part of the returning value of the function (**ID**) for the purpose of identifying the subject.

type *numeric(1), 1,2,...,7*; specifies the type of the slide. Details as below,

1. Time restricted display that does not allow the subject to enter any answer (that is, for display information only) and that information is displayed for a limited amount of time controlled by `time`.
2. Time unrestricted display that does not allow the subject to enter any answer (like `type=1`) but the information is display until the subject click on the green button located in the bottom of the slide to proceed to the next slide.
3. Numerical response display that allows the subject to give a numerical answer by clicking on the virtual keyboard. The location of the virtual keyboard is controlled by `box.level`.
4. Multipe choice response display that allows that subject to choose one or more selections from the given ones. Choices are controlled by `choices` while the number of selection is controlled by `n.select`. Also see `layout`.
5. Sequential Likert-scale response display that allows the subject to choose one of the Likert items. It is sequential in the sense that `cond` may contain multiple items where the similar questions are repeated asked but with different conditions (`cond`).
   For example: "The price is 10", "The price is 20", "The price is 30" are asked. Here `cond=seq(10,30,10)`. The Likert items are controlled by `scale`.
6. "Number line marking" response display allows the subject to give a perceptual numerical response (or multiple response) by clicking on the numberline (Kanayet & Opfer 2009, Siegler & Opfer 2003). The subject can only see a mark (or marks if multiple) on the number line, not the number it represents. `range` controls the length and the tick marks of the number line and `grid` controls the number of
   minor lines drawn between any two tick marks.
7. Gabor-Granger method (Gabor & Granger 1966, 1979) display that asks sequential questions in regarding to the acceptance of prices in an ascending or descending order. The order is controlled by `order`. `initial` controls the first value to be asked and subsequent value
   will increase or decrease by `increment`. Subjects can choose either "Yes" or "No" and the procedure ceases when "No" is answered.

`title` *character(1)*; the title to be displayed on the top of the slide. Should be made as short as possible since it will appear in one of the returning values **Question.number**. In fact, it is recommended to use titles like "Question 1" and "Demopgrahics 1" etc.

`content` *character()*; texts to be displayed in the slide. It can be a vector `c( )` containing multiple items. Each item must be quoted. **Note: for** `type=5` **or** `7` **the first component becomes the question that** `cond` **is attached to.** `cond` **is placed after** `content`.

`h` *numeric(), [0,100]*;the horizontal location of `content` hence its length must be the same as that of `content`. Must be in the interval [0,100].

`v` *numeric(), [0,100]*;the vertical location of `content` hence its length must be the same as that of `content`. Must be in the interval [0,100].

`size` *numeric()*; the font size of `content` hence its length must be the same as that of `content`.

`col` *character()*; the colour of `content` hence its length must be the same as that of `content`.

`time` **(only applies to** `type=1`**)**; *numeric(1)* the amount of time (in seconds) that the slide is displayed.

`interval` **(only applies to** `type=3`**)**; *numeric(2)* the permitted interval the numerical response can fall within.

`isinteger` **(only applies to** `type=3`**)**; *logical(1)* `TRUE` if the numerical response is only allowed to be an integer.

`choices` **(only applies to** `type=4`**)**; *character()* or *numeric()* the choices the subject can choose from.

`layout` **(only applies to** `type=4`**)**; *numeric(2)* the layout (number of rows and columns respectively) of the choices.

`n.select` **(only applies to** `type=4,6`**)**; *numeric(1)* the number of choices can be selected if `type=4` or the number of selection can be made if `type=6`.

`box.level` **(only applies to** `type=3,4,5,6,7`**)**; *numeric(2), [0,100]* if `type=3` defines the vertical location of the top of the first box of the choices.
numeric(1), [0,100] if `type=4,5,6,7` defines the horizontal and vertical location of the virtual keyboard. Must be between 0 & 100.

`cond` **(only applies to** `type=5`**)**; *character()* or *numeric()* a vector of conditions asked in the sequential Likert-scale question. Example: if `content=c("The price is","Please choose one of the following")`
and `cond=seq(5,15,5)` then three slides will display each of the followings: "The price is 5", "The price is 10" and "The price is 15" plus "Please choose one of the following" in each of these slides. **Note: the first component of** `content` **becomes the question that** `cond` **is attached to.** `cond` **is placed after** `content`.

`scale` **(only applies to** `type=5`**)**; *character()* or *numeric()* a set of Likert items the subject can choose from.

`grid` **(only applies to** `type=6`**)**; *numeric(1)* the number of minor grids lines to be drawn in between any two tick marks in `range`.

`range` **(only applies to** `type=6`**)**; *numeric(3)* a set of numbers to be displayed as ticks on the number line. The maximum and the minimum become the two ends of the number line.

`unit` **(only applies to** `type=6`**)**; *character(1)* or *numeric(1)* to be displayed as the unit (such as "$", "%", etc.) of the figures on the number line. This is placed right underneath the number line.

`initial` **(only applies to** `type=7`**)**; *numeric(1)* the starting value serves as the condition attached to the first sequential question in Gabor-Granger method. Subsequent conditions will decrease (if `order="descending"`) or increase (if `order="ascending"`) with the increment of `increment`.

`increment` **(only applies to** `type=7`**)**; *numeric(1)* the amount that the condition increases or decreases by (see `initial` above).

`order` **(only applies to** `type=7`**);** `"ascending"` *or* `"descending"` It defines increasing or decreasing conditions attached to the sequential question.

`Extra` **See Details below for usage and details**
`list( list(), list() ,...)` A list (with usage: `Extra=list()`) contains another sub-list or multiple sub-lists. Each sub-list refers to a table or a diagram to be displayed hence multiple tables or diagrams are allowed.

`Line.draw` **See Details below for usage and details**
Any quoted R commands in the form of character( ) (example: `c("abline(v=0, col='red')","lines(seq(-10,20,0.01),seq(-10,20,0.01)*1.2+100, col='yellow')")`) These commands will be executed. The aim is to provide further adjustments to the slide. **Note that single-quotes ' ' should be used to avoid the confusion with the double-quotes " ".**

Details

Extra

This argument aims to create plots and tables in the slide hence `Extra` is one of the optional arguments in `slide`. The usage of this argument must follow the list containing sub-lists format: `slide(... ,Extra=list( list(), ... ), ...)`.
For example, `Extra=list(list(...,type="p",...),list(type=...,"p",...) ,list(type=...,"t",...))` refers to two point plots and a table to be created while `Extra=list(list(type="h2",...)` while refers to a relative-frequency histogram to be created.
The "list containing sub-lists" format must always be used regardless of the number of plots and tables.
What arguments are required in each sub-list depend on `type`.
The below table (in a few partitions) gives a summary,

type=	loc	sub-list[[1]]	sub-list[[2]]
p (Point plot)	Required	Required	Required
l (Line plot)	Required	Required	Required
b (Both point & line plot)	Required	Required	Required
h1 (Histogram with frequency)	Required	Required	
h2 (Histogram with relative frequency)	Required	Required	
t (Table)	Required		
Input type:	numeric(4)	numeric() or character()	numeric()

type=	ticks	xlabl	ylabl
p (Point plot)	Required	Optional	Optional
l (Line plot)	Required	Optional	Optional
b (Both point & line plot)	Required	Optional	Optional

h1 (Histogram with frequency)	Required	Optional	Optional
h2 (Histogram with relative frequency)	Required	Optional	Optional
t (Table)			
Input type:	list(2) if type="p","l","b" or list(1) if type="h1","h2"	character(1)	character(1)

type=	title	col	table	size
p (Point plot)	Optional			
l (Point plot)	Optional			
b (Both point & line plot)	Optional			
h1 (Histogram with frequency)	Optional	Optional		
h2 (Histogram with relative frequency)	Optional	Optional		
t (Table)			Required	Required
Input type:	character(1)	character()	matrix()	numeric(3)

- loc: *numeric(4)* the location of the plot or table. These four numbers defines the left, lower, right and upper bound of the plot or table.
- sub-list list[[1]]: *numeric() or character()* The first component of the sub-list. If type="p", "l" or "b", this is the data shown on the x-axis. If type="h1" or "h2", this is the data whose frequency or relative frequency will be evaluated.
- sub-list list[[2]]: *numeric()* The second component of the sub-list, this is the data shown on the y-axis.
- ticks: *list(2) if* type="p","l","b" *or list(1) if* type="h1","h2" For type="p","l" or "b", ticks[[1]] and ticks[[2]] become the tick marks of the x- and y-axis respectively. For type="h1" or "h2", ticks[[1]] becomes the tick marks on the y-axis.
- xlabl: *character()* The lable of the x-axis. The name attached to list[[1]] will be used instead.
- ylabl: *character()* The lable of the y-axis. The name attached to list[[2]] will be used instead.
- title: *character()* The title of the plot.
- col: *character()* The colour of each bar for type="h1","h2". The bars are not coloured (or white) if NULL.
- table: *a matrix with* dimnames( ). The table will be created with the dimension and, with the row & column names the same as that of table.
- item: *numeric(3)* refers to the font size of the row, column names then in-table information.

Line.draw
A single or a set of double quoted R commands. Any texts in between the quotes will be executed. A vector `c( )` is used for multiple commands.
Since the double quotes are reserved in this case, single quotes should be used in the commands instead. For example, `Line.draw="abline(v=0,col=\bold{'}red\bold{'})"`
instead of `Line.draw="abline(v=0,col=\bold{"}red\bold{"})"`.

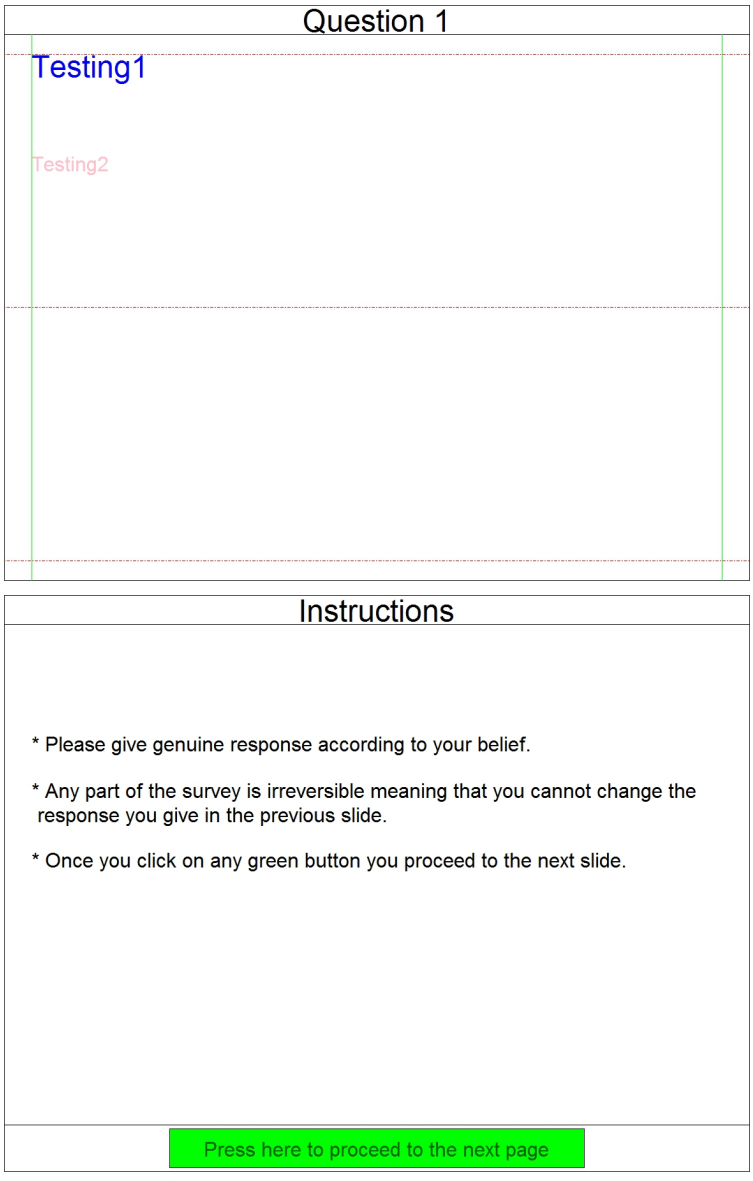
Value

A 5-column matrix is returned in the R console. The column names are **ID**, **Question.number**, **type**, **Condition.Likert** and **Response**.

- **ID** shows `id`.
- **Question.number** is `title` but without spaces.
- **type** is `type`, the slide type (a integer between 1 and 7).
- **Condition.Likert** is the `cond` in the sequential Likert-scale question, shown as `NA` if `type` is not 7.
- **Response** is the response given by the subject. Therefore, the number of rows may vary depending on number of the response. The returning values of each slide type is described below,

  1. No values returned.
  2. No values returned.
  3. The numerical response given by the subject by clicking the virtual keyboard upon the restrictions of `interval` and `isinteger`.
  4. A number representing the choice selected (i.e. "1" refers to the first choice being selected, "2" to the second, ...). The number of rows then depends on how many choices the subject is allowed to select as controlled by `n.select`.
  5. The exact choice (one of the items in `scale`) the subject selects. That is, if `scale` is of character() so is the response. The number of rows depends on the number of conditions being asked (i.e. length of `cond`). Also see **Condition.Likert** above for another returning value.
  6. A number that the mark (or marks if multiple selections allowed) represents on the number line. The number of rows then depends on how many selections (i.e. marks) the subject is allowed to make as controlled by `n.select`.
  7. The number (the price in the Gabor & Granger 1966 1979) asked in the sequential question in which the subject answers "No".

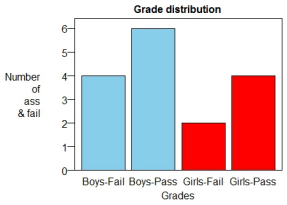
Returning graphics of examples



Question 2

What is the average score of all groups?

7	8	9
4	5	6
1	2	3
0	.	del
enter		



	Group 1	Group 2	Group 3	Group 4
Gender	M	F	M	F
Average Score	89.36	105.78	90.5	82.18
Test type	Test A	Test A	Test B	Test B

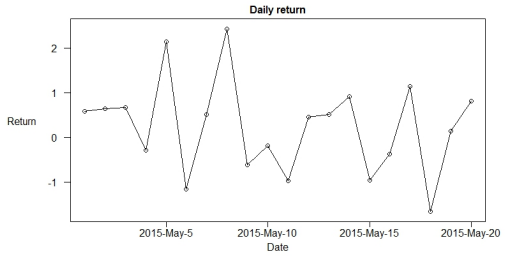
Question 3

What do you think the return on the 2015-May-21 will be?

Please choose one of the followings

Goes down | Remains the same | Goes up

Only one selection is permitted enter delete



Question 4

The market price is $ 10

(0 refers to 'Strongly disagree' ... 5 refers to 'Strongly agree')

Single click one of the options

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

Question 5

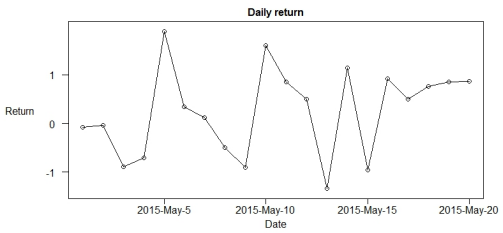
Within which two numbers do you think the return on the 2015-May-21 will fall in

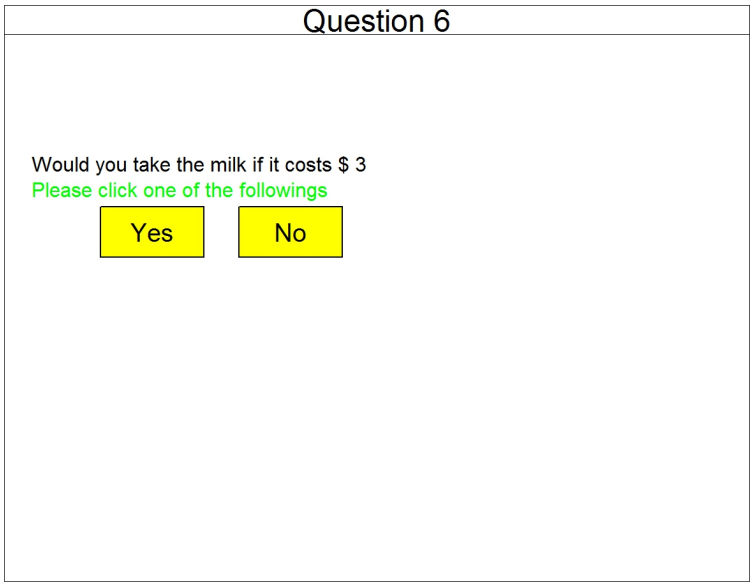
To select, place the cursor on the below measurement tape and single click the r

-3 -2 -1 0 1 2 3

in %

Submit selection(s) Delete





Author(s)

Char Leung

References

Gabor, G & Granger, CWJ (1966) "Price as an indicator of Quality: Report on an Enquiry", *Economica*, **33**(129), pp.43-70.
Gabor, G & Granger, CWJ (1979) "The Attitude of the Consumer to Prices", *Management Decision*, **17**(8), pp.619-634.
Kanayet, F & Opfer, J (2009) "Why Children's Number-line Estimates Follow Fechner's Law", *Cognitive Science Conference Proceedings 2009*, pp.1936-41.
Siegler, RS & Opfer, JE (2003) "The Development of Numerical Estimation: Evidence for Multiple Representations of Numerical Quantity", *Psychological Science*, **14**(3), May 2003, pp.237-243.

See Also

cover,identity,read.write

Examples

```
## Not run: slide(id=1234,type=1,title="Question 1",content=c("Testing1","Testing2"),
h=c(0,0),v=c(100,80),size=c(3,2),col=c("blue","pink"),time=4,Line.draw=c("abline
(h=seq(0,100,50),col='red',lty=4)","abline(v=seq(0,100,100),col='green')"))
## End(Not run)
## Not run: slide(id=1234,type=2,title="Instructions",content=c("* Please give
genuine response according to your belief.","* Any part of the survey is
irreversible meaning that you cannot change the\n response you give in the previous
```

```
slide.","* Once you click on any green button you proceed to the next slide."),
h=c(0,0,0),v=c(80,70,55),size=c(2,2,2),col=c("black","black","black"))
## End(Not run)
## Not run: slide(id=1234,type=3,title="Question 2",content="What is the average
score of all groups?",h=0,v=100,size=2,col="black",box.level=c(70,100),Extra=list
(list(date=c(rep("Boys-Pass",6),rep("Girls-Pass",4),rep("Boys-Fail",4),
rep("Girls-Fail",2)),type="h1",loc=c(10,30,40,60),ticks=list(seq(0,6,1)),
xlabl="Grades",ylabl="Number\nof\nass\n& fail",title="Grade distribution",
col=c(rep("skyblue",2),rep("red",2))),list(Table=matrix(c(rep(c("M","F"),2),
round(rnorm(4,100,10),2),rep("Test A",2),rep("Test B",2)),ncol=4,byrow=T,
dimnames=list(c("Gender","Average\nScore","Test\ntype"),paste("Group",1:4))),
size=c(1.5,1.5,2),type="t",loc=c(55,30,90,60))))
## End(Not run)
## Not run: slide(id=123,type=4,title="Question 3",content=c("What do you think
the return on the 2015-May-21 will be?","Please choose one of the followings"),
h=c(0,0),v=c(90,80),size=c(2,1.5),col=c("black","blue"),choices=c("Goes down",
"Remains the same","Goes up"),layout=c(1,3),box.level=75,n.select=1,
Extra=list(list(Date=paste("2015-May-",1:20,sep=""),return=rnorm(20),type="b",
loc=c(20,10,80,50),ticks=list(c(seq(0,20,5)),c(seq(-2,3,1))),ylabl="Return",
title="Daily return")))
## End(Not run)
## Not run: slide(id=123,type=5,title="Question 4",content=c("The market price is
$","(0 refers to 'Strongly disagree' ... 5 refers to 'Strongly agree')"),
h=c(0,0),v=c(80,70),size=c(2,1.5),col=c("black","orange"),cond=seq(10,12,0.5),
scale=seq(0,5,0.5),box.level=50)
## End(Not run)
## Not run: slide(id=123,type=6,title="Question 5",content="Within which two numbers
 do you think the return on the 2015-May-21 will fall into?",h=0,v=100,size=2,
col="black",range=seq(-3,3,1),grid=5,unit="%",box.level=85,n.select=2,Extra=
list(list(Date=paste("2015-May-",1:20,sep=""),return=rnorm(20),type="b",loc=
c(20,10,80,45),ticks=list(c(seq(0,20,5)),c(seq(-2,3,1))),ylabl="Return",title=
"Daily return")))
## End(Not run)
## Not run: slide(id=123,type=7,title="Question 6",content="Would you take the milk
if it costs $",h=0,v=80,size=2,col="black",initial=3,increment=1,order="descending"
,box.level=70)
## End(Not run)
```

Index