Go-No Go Tutorial

Milla Pihlajamaki

27 5 2022

Go-No Go Task Tutorial

This tutorial will demonstrate how the *gonogo* package is used. The package contains two functions: play_gonogo() for playing the Go-No Go Task, and check_rt() for checking for irregularities in the output data, specifically the reaction time column.

Go-No Go Task in Short

The Go-No Go Task is a widely used test to measure inhibitory control, a cognitive process that enables humans to cancel motor activity after its initiation. It requires the participant to perform an action given certain stimuli (Go stimuli), and inhibit that action under a different set of stimuli (No Go stimuli).

There are two parameters in the experimental design that are especially important: the length of each trial and the relative proportion of the Go and No-Go trials. Fortunately, both these parameters can be easily manipulated in the gonogo() function: length of the trial with the *inter* argument, and the relative proportion of the Go and No-Go trials with the *prb* argument.

In addition to these two arguments, you can specify the participant *id* (name or unique id number), *n_trial* (number of trials), *n_block* (number of blocks), and *stimuli* (the Go and No Go stimuli).

How to Use the play_gonogo() Function

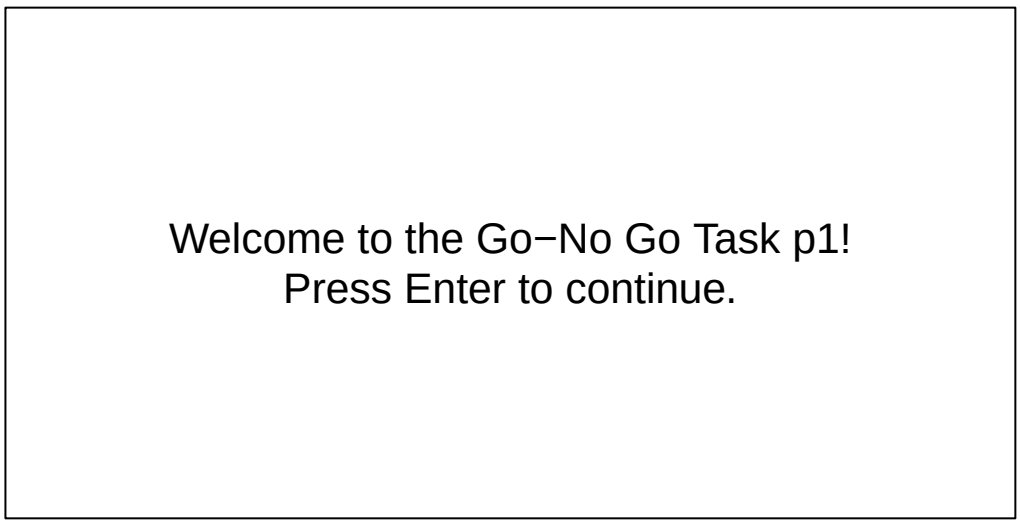
The following code gives an example of how the play_gonogo() function can be used.

```
# Load package
# devtools::install_github("Programming-The-Next-Step-2022/GoNoGoTask/gonogo")
library(gonogo)
```

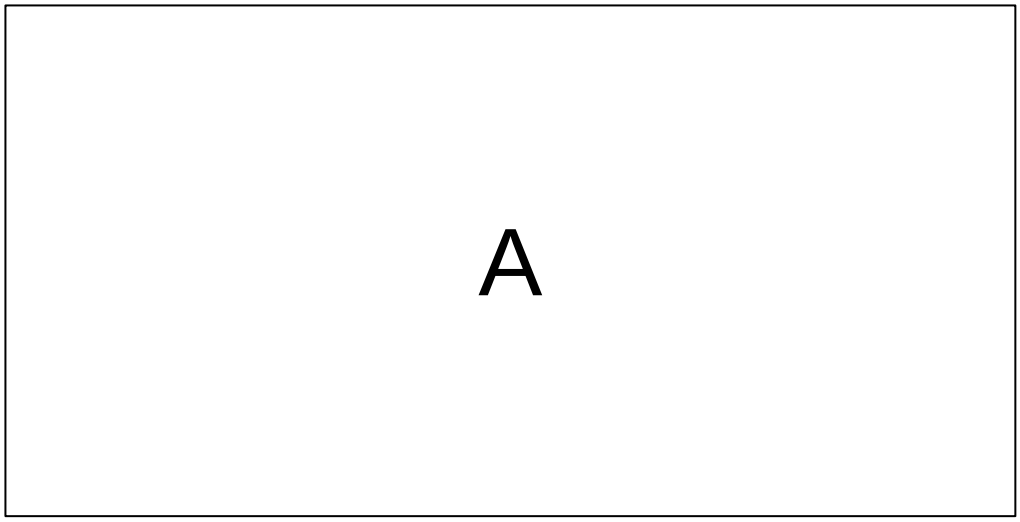
```
# Run the Go-No Go Task and save the output in an object
p1_data <- play_gonogo(id = "p1",  # id
                       n_trial = 5,  # number of trials
                       n_block = 3,  # number of blocks
                       stimuli = c("A", "X"),  # Go and No Go stimuli
                       inter = 0.6,  # interval (length of trials)
                       prb = c(0.8, 0.2))  # relative proportion
```

Example Screens from the Task

Below you can find examples of how the task looks: the welcome screen, information about the number of trials and blocks, and finally an example of how the stimuli are presented during the task.



The task consists of 3 blocks, each consisting of 5 trials.
There is is a break between each pair of blocks.
Before starting the task, there is one practice block of 10 trials.
Press Enter to start the practice trials.



What the Output Data Looks Like

```
##    id response correct              SDT          rt stimulus block
## 1  p1     none       1 correctrejection          NA        X     1
## 2  p1     none       1 correctrejection          NA        X     1
## 3  p1    space       1              hit  0.47666382        A     1
## 4  p1    space       1              hit  0.55146193        A     1
## 5  p1     none       0             miss          NA        A     1
## 6  p1    space       0       falsealarm  0.56809711        X     2
## 7  p1     none       0             miss          NA        A     2
## 8  p1    space       1              hit -0.01646210        A     2
## 9  p1    space       1              hit  0.59251689        A     2
## 10 p1     none       1 correctrejection          NA        X     2
## 11 p1     none       0             miss          NA        A     3
## 12 p1    space       1              hit -0.01505090        A     3
## 13 p1     none       0             miss          NA        A     3
## 14 p1    space       1              hit -0.02203418        A     3
## 15 p1     none       0             miss          NA        A     3
```

How to Read the Output

The play_gonogo() function returns a dataframe consisting of n_trial*n_block (number of trials times number of blocks) rows and seven columns:

id = participant's name or id as specified

response = response key used on the trial (space when participant responded, none when no response was given)

correct = whether the response was correct or not (1=correct, 0=incorrect)

SDT = responses categorized according to Signal Detection Theory; click here to read more about Signal Detection Theory

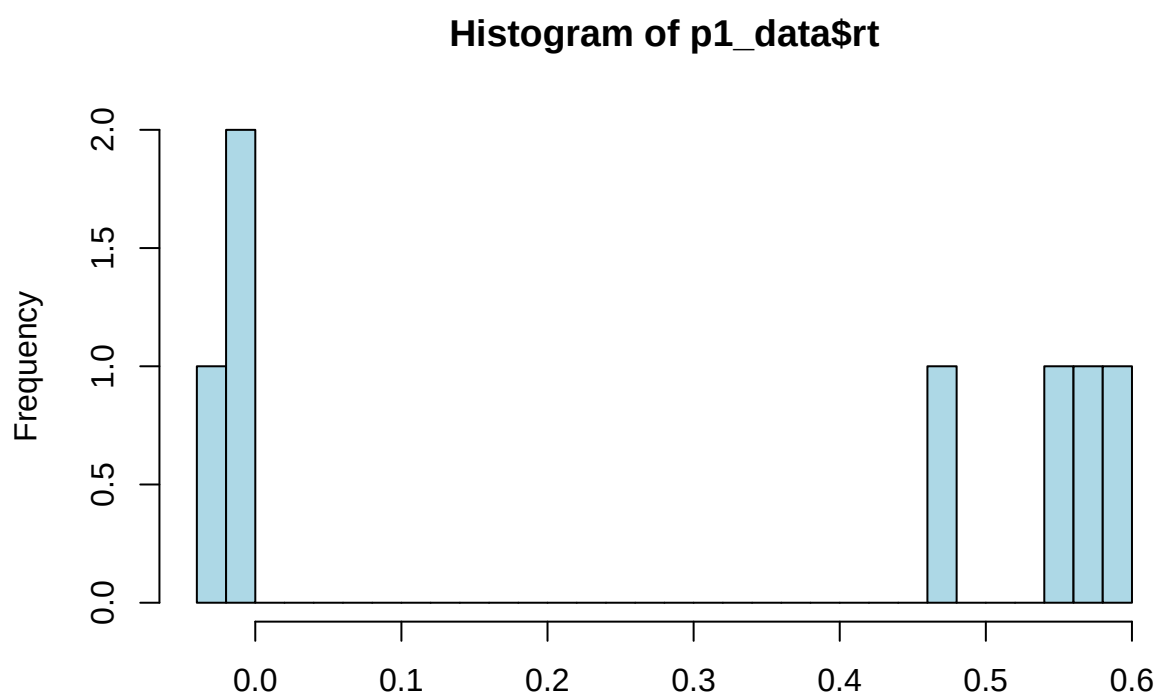
rt = reaction time in seconds (NA when participant did not respond during that trial)

stimulus = the stimulus shown on the trial

block = the block number

The check_rt() Function

It appears that there is something weird going on in the reaction time column:

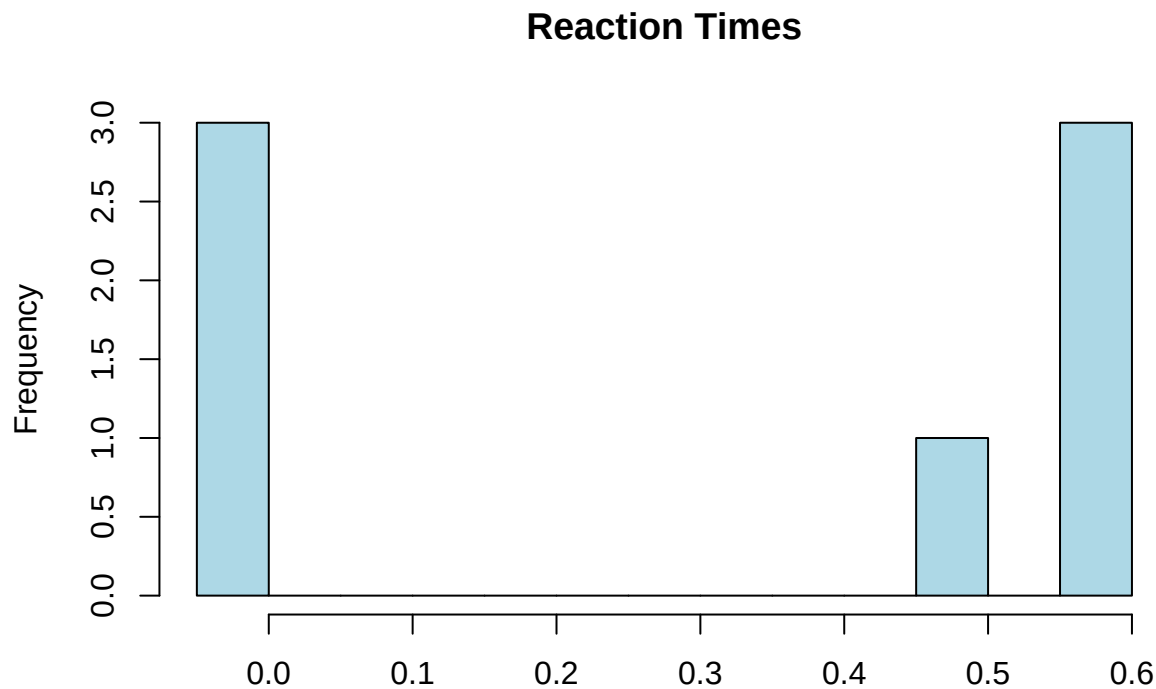


There are some observations that are very close to (or below) 0. This is the case when the participant takes too long to respond to a stimulus, and their response "leaks" into the next trial. As the function corrects for the time that it takes to run the function to gain more accurate reaction times by subtracting a constant from the reaction times, this might sometimes lead to very small or negative values.

Fortunately, the gonogo package comes with a function that helps spot irregularities in reaction times: check_rt()! The check_rt() function points our unsually long distances between sorted observations in your reaction time data. It has two arguments, *data* (specify your dataframe), and *ratio* (the ratio of the longest distance between sorted data points and the range of the data). See below:

```r
check_rt(data = p1_data, ratio = 1/3)
```

```
## [1] "There are observations in your reaction times that are unusually far from each other: the ratio
```



End of the Tutorial!