



Contents lists available at ScienceDirect

## Journal of Behavioral and Experimental Finance

journal homepage: [www.elsevier.com/locate/jbef](http://www.elsevier.com/locate/jbef)

Full length article

oTree: The “bomb” risk elicitation task<sup>☆</sup>Felix Holzmeister<sup>a,\*</sup>, Armin Pfurtscheller<sup>b</sup><sup>a</sup> Leopold-Franzens Universität, Department of Banking and Finance, Universitätsstraße 15, Innsbruck, Austria<sup>b</sup> Armin Pfurtscheller Webdesign & Programmierung, Stackler 58, 6167 Neustift, Austria

## ARTICLE INFO

## Article history:

Received 23 February 2016

Received in revised form 4 March 2016

Accepted 8 March 2016

Available online 26 April 2016

## Keywords:

oTree

Bomb risk elicitation task

Risk preferences

## ABSTRACT

Among recently suggested elicitation procedures for risk preferences, the “bomb” risk elicitation task (BRET) by Crosetto and Filippin (2013) gained noticeable attention. This article presents a ready-to-use software module for use with oTree (Chen et al., 2015) which allows for easily conducting the BRET in numerous different variants, tapping the potential and advantages of state-of-the-art web programming technologies oTree is based on. In a user-friendly and straightforward way, predefined and thoroughly documented variables are specified in a separate file, providing full control over different aspects of game play. Furthermore, the application is prepared for multilingual use, providing a convenient way for translation into different languages. Additionally offered *Stata* script files facilitate data processing and the evaluation of individual level risk preferences.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Risk is prevalent in numerous domains of economic decision-making. Investigating the implications of individual attitudes towards risk and uncertainty on different aspects of economic behavior has been on the agenda of experimental scholars for several decades. A still growing body of literature proposes an extensive set of tools and methods for properly measuring risk attitudes, commonly used in many fields of economic research. Among recently suggested elicitation procedures for risk preferences, the “bomb” risk elicitation task (BRET) put forward by Crosetto and Filippin (2013) gained noticeable attention. As an intuitively comprehensible elicitation procedure providing a

fine-grained measure of individual level risk attitudes, the BRET is an up-and-coming alternative to well-established methods commonly applied in experimental research.

This article presents a ready-to-use software module for use with oTree (Chen et al., 2015) which allows for easily conducting the BRET in different variants proposed by the original authors as well as some natural extensions. As an open-source, object-oriented environment, oTree provides a platform-independent software solution deployable on any device such as smartphones or tablet computers. By this means, oTree enables the implementation of experiments in the field, the laboratory, or online, as well as any combination thereof. As a ready-made oTree application, our software module for conducting the BRET utilizes all the advantages of oTree, implying straightforward setup and usage, state-of-the-art graphical display, seamless integration into existing programs, and enhanced flexibility.

## 2. The bomb risk elicitation task

The bomb risk elicitation task (BRET) is a visual (real-time) elicitation method for risk preferences requiring subjects to decide on how many out of a matrix containing

<sup>☆</sup> The software application described here is free of charge and licensed under the MIT open source license with a citation requirement. Any use of the software – whether as a whole or in parts – implies the acceptance of the license agreement available in the folder for download. Demo-versions of several task variants are available at [bret-demo.herokuapp.com](http://bret-demo.herokuapp.com). The application is available for download at the website of Felix Holzmeister ([www.holzmeister.biz](http://www.holzmeister.biz)) and at [https://github.com/JBEF/oTree\\_BRET](https://github.com/JBEF/oTree_BRET).

\* Corresponding author.

E-mail address: [felix.holzmeister@uibk.ac.at](mailto:felix.holzmeister@uibk.ac.at) (F. Holzmeister).

URL: <http://www.holzmeister.biz> (F. Holzmeister).

$m \times n$  boxes to collect, one of which hides a bomb. Each box collected pays off the same amount of money ( $\gamma$ ) such that potential earnings increase linearly. However, earnings are zero if the bomb is contained in one of the collected boxes. By this means, for some given  $m$ ,  $n$ , and  $\gamma$ , the BRET elicits consistent decisions in  $(mn + 1)$  lotteries – fully described in terms of outcomes and probabilities – by a single parameter, the number of boxes collected (Crosetto and Filippin, 2013).

Subjects' decisions can be formalized as the choice of Lottery  $L_k$ , where  $k$  denotes the number of boxes collected and  $m$  and  $n$  refer to the number of rows and columns, respectively, such that

$$L_k = \begin{cases} 0 & \text{with prob. } p = k/mn \\ \gamma k & \text{with prob. } p = (mn-k)/mn. \end{cases} \quad (1)$$

The expected value of any lottery  $L_k$  is  $E(L_k) = \gamma(k - k^2/mn)$  which is trivially equal to zero for  $k = 0$  and  $k = mn$  and attains a maximum at  $k = mn/2$ . Assuming a power utility function of the form  $u(x) = x^r$ , a subject maximizes his or her utility by stopping the collecting process after  $k^* = mn \cdot r/(1+r)$  boxes.

Compared to other commonly used risk elicitation methods such as multiple choice lists between pairs of lotteries with varying probabilities (Hey and Orme, 1994; Holt and Laury, 2002), multiple decisions between a fixed lottery and varying certain payments (Dohmen et al., 2010; Abdellaoui et al., 2011), or single choice problems of how much to allocate between a safe and a risky asset (Gneezy and Potters, 1997; Charness and Gneezy, 2010), the BRET captivates by its simple and intuitive design, requiring minimal numeracy skills, and by allowing for a precise elicitation of individual-level risk preferences. By reconciling simplicity and fine-grained measurements of risk preferences, the BRET is well-placed in the trade-off between comprehensibility and precision discussed by Dave et al. (2010) and Charness et al. (2013). Moreover, the BRET neither suffers from the impact of loss aversion – as it does not provide any endogenous reference point outcomes could be compared to – nor from inconsistent decisions as it entails a unique choice (Crosetto and Filippin, forthcoming).

Software modules for conducting the BRET are available for download on Paolo Crosetto's webpage, either as a z-Tree version (Fischbacher, 2007) or as a stand-alone Python application. However, while the former can only be run under Windows operating systems and might be subject to network problems for some treatment variations, the latter is a local client application requiring that Python – together with a graphical user interface library – is installed on any computer the task is intended to run on. Furthermore, both applications are restricted to specific pre-determined treatments only.

Our application alleviates any of these drawbacks by utilizing state-of-the-art web programming technologies and adds further advantages, especially increased flexibility, straightforward setup of the task and unrestricted combinations of any variations described below. As a ready-to-use oTree app (Chen et al., 2015), our module taps the full potential of oTree and, thereby, is qualified to be run on any operating system and any device—be that a

smartphone, a tablet, or a desktop computer—in a state-of-the-art and graphically appealing manner. The only requirement for running the task is an arbitrary web browser being installed and JavaScript being enabled.<sup>1</sup>

### 3. Setup and usage

The BRET software module is programmed in JavaScript using the AngularJS framework by Google and is seamlessly embedded within a stand-alone oTree application. Therefore, the app can be used in any experiment conducted with the oTree framework by copying the app folder into oTree's Django project directory and altering the session configurations in oTree's settings.py file.

The application allows experimenters to implement the BRET in several different variants. In a user-friendly and straightforward way, predefined and thoroughly documented variables are specified in a separate file (setup.py) at the root of the application directory. The file setup.py consists of the Constants class of the oTree app and contains several variables to be specified in order to determine game play. Since the app is programmed as a standard oTree application, the range and scope of existing variables can easily be altered or extended by custom-designed functions.

The entire application is prepared for multilingual use utilizing Django's i18n internationalization routines. Instructions on how to apply the translation features are included in the folder for download.

#### 3.1. General specifications and appearance

Several configurations in the file setup.py allow for defining general task properties. The decimal field box\_value determines the (potential) payoff for each box collected. The location of the bomb is determined randomly as soon as the task is loaded. While the original experiment by Crosetto and Filippin (2013) consists of a  $10 \times 10$  grid of boxes – underpinned by the argument that probabilities are most easily recognized given a total of 100 boxes – our application allows for conducting the task with any arbitrary none-prime number of boxes by simply specifying the number of rows and columns (num\_rows and num\_cols in setup.py, respectively).<sup>2</sup>

As for any standard oTree application, the field num\_rounds can be assigned any positive integer to determine the number of rounds to be played. For the case of num\_rounds larger than one, the variable random\_payoff may be set to either True or False in order to define whether one round is randomly chosen

<sup>1</sup> Note that ad blockers or other browser-integrated filter mechanisms restricting the functionality of JavaScript may prevent the app to load correctly. As users cannot be compelled to accept JavaScript applications, this should be considered when running the task in a non-controlled environment.

<sup>2</sup> In addition, the size of the boxes to be displayed can be determined by setting the width and height of the boxes in pixels. This allows the researcher to ensure that the task can be displayed correctly on any device such as tablets or smartphones.

Download English Version:

<https://daneshyari.com/en/article/931678>

Download Persian Version:

<https://daneshyari.com/article/931678>

[Daneshyari.com](https://daneshyari.com)