Implementing a Holt and Laury (2002) Multiple Price List lottery in oTree

Olaf Ghanizadeh

What is oTree?

What is oTree?

- Open Source framework to create behavioral economics experiments, Chen, Schonger, and Wickens (2016)
- Python based, uses Django to create a web application
- Bootstrap 4 for styling
- Encourages object-oriented programming, extensive use of classes and methods in order to avoid code duplication

Basic Strucutre

```
oTree
├── settings.py
└── hl_mpl
    ├── templates
    │   └── hl_mpl
    │       └── template files
    ├── config.py
    ├── models.py
    ├── pages.py
    └── utils.py
```

How does oTree work?

```
1     class Subsession(BaseSubsession):
2     """Create the session to be played"""
3       n = self.session.config['num_choices']
4       # Store in session variables
5       self.session.vars['probs'] = probs
6       def creating_session(self):
7       """Session variables that holds for all players in this session"""
8         for p in self.get_players():
9         """Each player, p, in the session, self"""
10
11    class Group(BaseGroup):
12      """If we wanted to have players in different  groups"""
13        pass
```

In `models.py` we execute most of our code:

```
1       class Player(BasePlayer):
2       """We execute what goes for each player."""
3         name = models.StringField()
```

We register the fields that will be exposed on the front-end, and the variables that will be stored in each 'player object'

```
1       def set_payoffs(self):
```

A method within the Player Class where we can set the payoff for each player. We need to call this method for payoffs to be set. In my project it is called before the user sees the final results page.

In pages.py we create the pages that will be shown to the user, define the forms, expose variables for the front end and so on. Each instance of Page gets linked to a HTML file in the templates folder.

```
1  # Class for the IntroPage. Inherits attributes from Page Class
2  class IntroPage(Page):
3      # Get forms to be displayed on IntroPage
4      form_model = 'player'
5      form_fields = ['name', 'risk']
```

We register the fields that will be exposed on the front-end, and the variables that will be stored in each 'player object'

```
# Class for the DecisionPage. Inherits attributes from Page Class
class DecisionPage(Page):
    form_model = 'player'

    ...

    # Triggers the function that set draws the payoff of the
    #user before the user is taken to the result page. This
    # should be changed if we were to make a game with several rounds.
    def before_next_page(self):
        self.player.set_payoffs()
```

We call the `self.player.set_payoffs()` before the next page is loaded to set the payoffs for the individual player.

Templates

- HTML Templates made dynamic with Django Template Language (DTL): `{{player.name}}`
- Extendable with JavaScript

Experiment

Experiment

- A simplified version of the experiment popularized in Holt and Laury (2002)
- An approach to elicit individual preferences for risk in various contexts
- Assumes that Constant Relative Risk Aversion is valid

Goal in our experiment

Does the participant choose differently from their indicated risk preferences?

- A risk-neutral user should consider the expected values of the different outcomes, and maximize it for each choice.
- In our experiment that is the sequence: AAA/BBBBB
- The number of safe choices (A), or the switching point, can be used to estimate the parameters of an individual utility function and the participant's risk preferences.

Table 1: Probabilities, choices and expected values

p	$1-p$	A_{High}	A_{Low}	B_{High}	B_{Low}	$E[A]$	$E[B]$	Difference
0.125	0.875	200	160	385	10	165.0	56.875	108.125
0.250	0.750	200	160	385	10	170.0	103.750	66.250
0.375	0.625	200	160	385	10	175.0	150.625	24.375
0.500	0.500	200	160	385	10	180.0	197.500	-17.500
0.625	0.375	200	160	385	10	185.0	244.375	-59.375
0.750	0.250	200	160	385	10	190.0	291.250	-101.250
0.875	0.125	200	160	385	10	195.0	338.125	-143.125
1.000	0.000	200	160	385	10	200.0	385.000	-185.000

Acknowledgements

Acknowledgements

- Simplified version of experiment
- Should record if the user decides to change their mind. I.e. choosing $A \to B \to A$ in a specific choice.
- Code error with > 9 choices.
- Should include several rounds and treatments
- Incentives are not compatible: Limited value of resulting data

Data Analysis

Choices of 20 participants

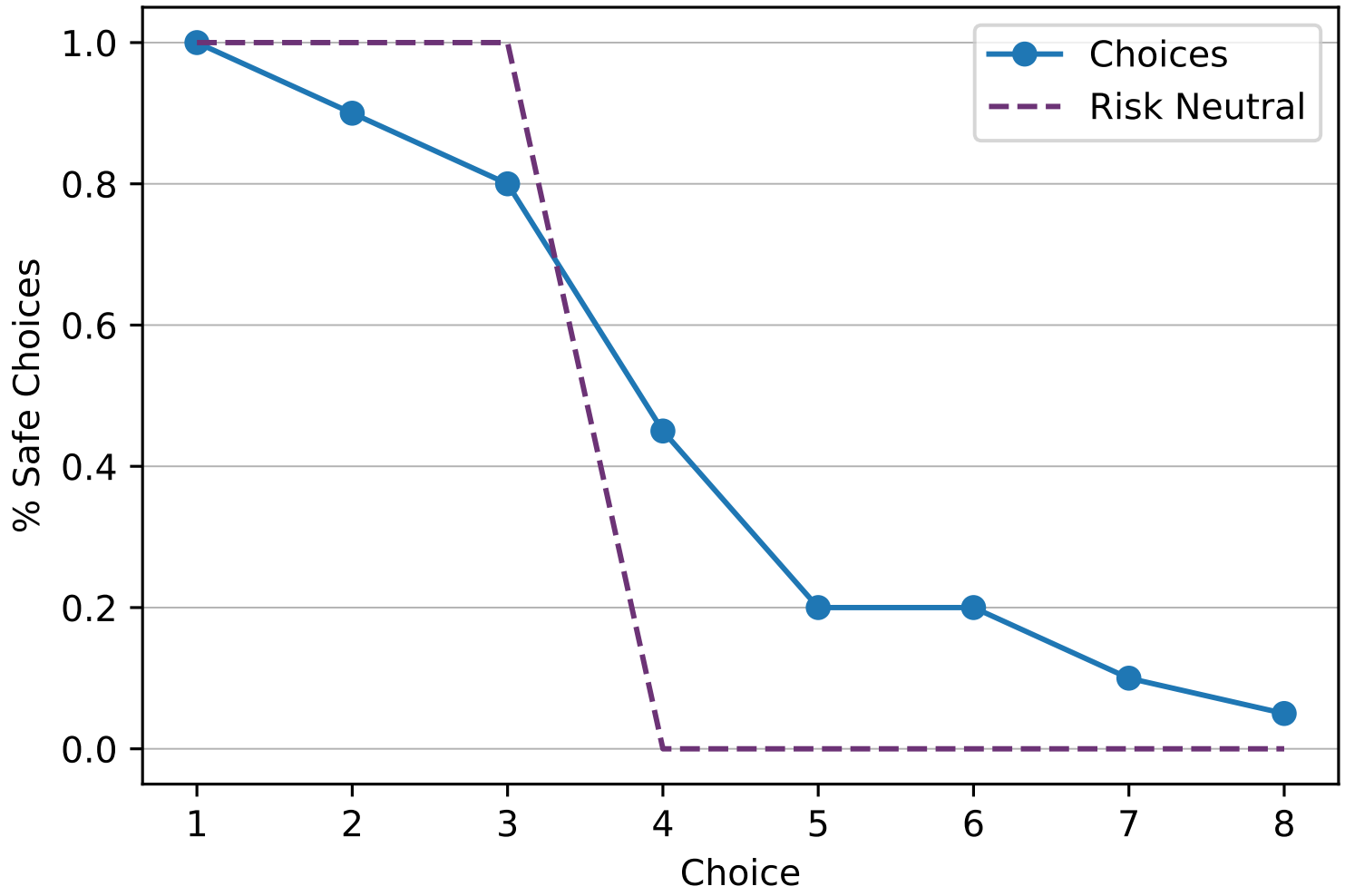

Figure 1: Choices made compared to a 'risk neutral' profile

A random sample

Indicated	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5	Choice 6	Choice 7	Choice 8
Risk Averse	A	A	A	A	A	A	B	B

Random sample visualized

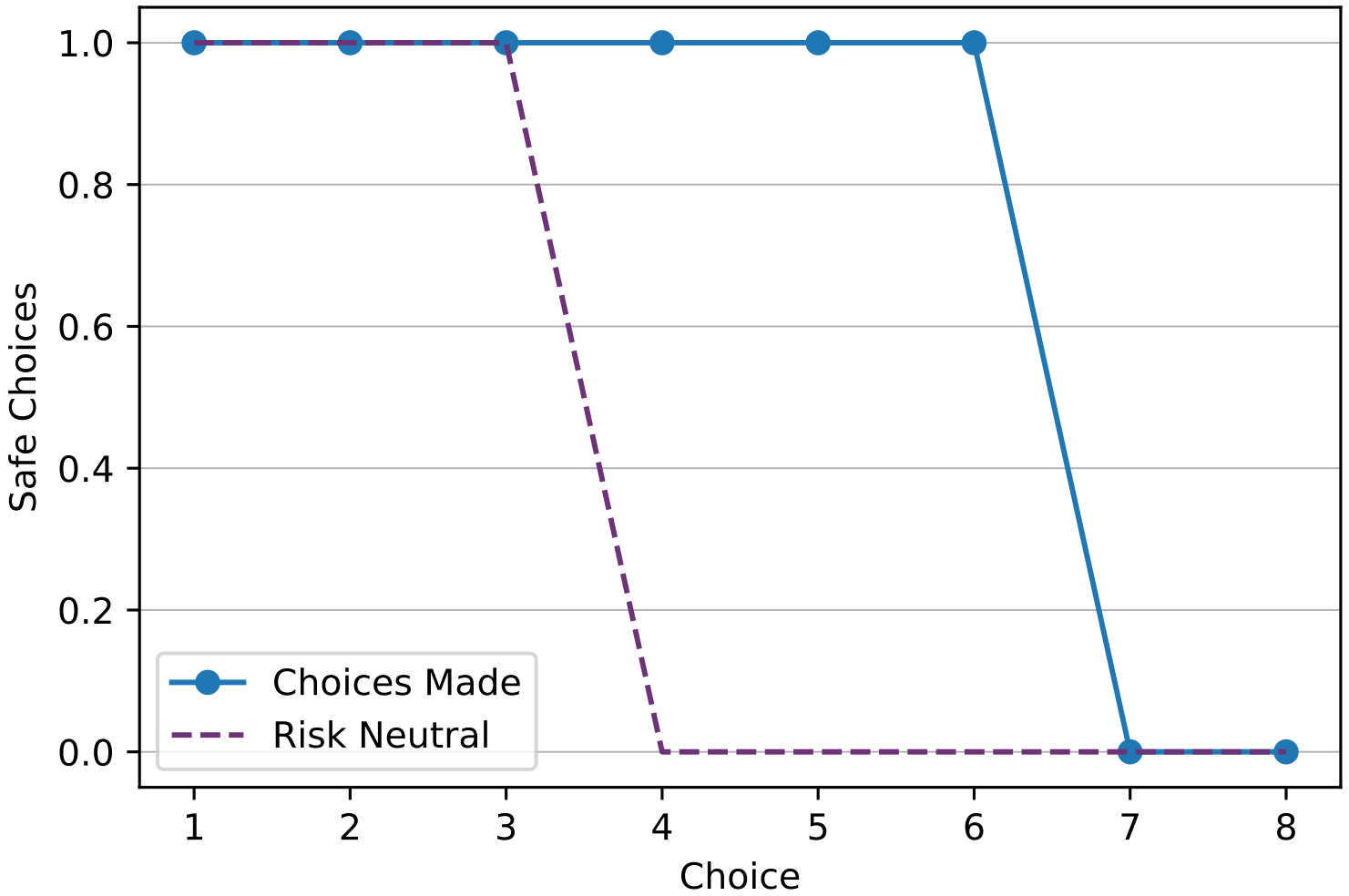

Figure 2: Choices made compared to the indicated profile

Flask Application

`https://progtech-flask-app.herokuapp.com/`

- Passes the data cleaned with Pandas to Flask
- Using d3.js for visualization
- Numpy to draw 2 random participants

Questions?

Experiment code: github.com/olafghanizadeh/hl_mpl

Project files: github.com/olafghanizadeh/progtech-project

References

- Chen, Daniel L., Martin Schonger, and Chris Wickens (Mar. 2016). "oTree—An Open-Source Platform for Laboratory, Online, and Field Experiments". In: *Journal of Behavioral and Experimental Finance* 9, pp. 88–97. ISSN: 22146350. DOI: 10.1016/j.jbef.2015.12.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S2214635016000101 (visited on 12/04/2019).
- Holt, Charles A and Susan K Laury (Nov. 2002). "Risk Aversion and Incentive Effects". In: *American Economic Review* 92.5, pp. 1644–1655. ISSN: 0002-8282. DOI: 10.1257/000282802762024700. URL: http://pubs.aeaweb.org/doi/10.1257/000282802762024700 (visited on 12/04/2019).