Workshop: Section 7 - Randomization and Template Variables

Philipp Chapkovski
University of Bonn
chapkovski@uni-bonn.de

September 11th - 12th, 2023

Quick reminder of overall oTree data structure

oTree Models: General Structure

- **Session**: Top-level container for the entire experiment.
  - Consists of participants.
  - Contains a sequence of apps.
- **App**: A component of the session.
  - Can have multiple rounds.
- **Round**: A single iteration within an app.
  - Divides players into groups or one large group (if `players_per_group=None`).
  - Includes all players in a subsession.
- **Group**: A subset of players within a round.
- **Subsession**: A set of all players in the round.
- **Player**: Individual subject in a round.
  - Each player has a corresponding **Participant**, which stays the same

Between-Session Randomization

- You can set session-specific parameters in SESSION_CONFIG to distinguish between different sessions.

Example SESSION_CONFIG:

```
dict(
    name='public_goods',
    app_sequence=['pgg'],
    num_demo_participants=3,
    random_endowment=True
)
```

- You can use these session-specific parameters in templates.

Example HTML Template:

```
Your endowment is {{ if session.config.random_endowment }} randomly assigned
{{ else }} 20 points {{ endif }}
```

creating_session

Within-Session Randomization: creating_session

- creating_session is a built-in function in oTree'
- It is executed once before the session starts
- The code runs **FOR EACH SUBSESSION**
- Commonly used for within-session randomization.

Example: Random Endowment for Each Player

```
from random import randint

def creating_session(subsession):
    for player in subsession.get_players():
        player.endowment = randint(10, 20)
```

Evenly Assigning Treatments Across Players

- You can use Python's itertools.cycle to iterate over a list indefinitely.

Example: Assigning 'fixed' and 'random' Treatments

```python
from itertools import cycle

def creating_session(subsession):
    treatments = cycle(['fixed', 'random'])
    for player in subsession.get_players():
        player.treatment = next(treatments)
```

You can do the same with groups (subsession.get_groups())

Using Variables in Templates

- How to dynamically display content in HTML templates using oTree's context variables.
- Introduction to basic Jinja2 template syntax for conditionals and loops.

JavaScript in Templates

- Brief overview of incorporating JavaScript for more dynamic user experiences.
- Examples of simple JavaScript functions and how to link them with oTree variables.

Exercise

- Participants will implement randomization in a simple game and use variables in templates.