2. Simulated Data

true

2022-10-29

Background

In an attempt to incorporate uncertainty to Gächter et al. (2017)'s dynamic public goods game (DPGG), I plan to run a series of remote online experiments using oTree (Chen, Schonger, and Wickens 2016). The first experiment will replicate Gächter et al.'s NOPUNISH 10-period version as close as possible (given the remote circumstances). The current demo version of the experiment can be found here. Click here to visit the corresponding Github repository.

This report is the second in a series of reports covering this project. It explains how simulated data is processed and prepared for analyses.

Replication Data

```
# DT <- read.csv("../../../data/simulation/all_apps_wide.csv",
#                stringsAsFactors = FALSE) %>% data.table()

files <- list.files(path = "../../../data/replication/",
                    recursive = TRUE,
                    full.names = TRUE)
csvs <- list()

for(i in files){
  name <- str_extract(string = i,
                      pattern = "2021.*")

  temp <- read.csv(i,
                   stringsAsFactors = FALSE) %>%
    data.table()

  csvs[[name]] <- temp
}

full <- rbindlist(l = csvs,
                  use.names = TRUE) %>%
  data.table()

DT <- full[participant._index_in_pages > 2 &
             participant.time_started != "" &
             participant.label != "" &
             !(is.na(dPGG.1.player.contribution))]
```

```
rm(list = c("files", "temp"))
```

Meta data

```
meta_long <- full[participant.time_started != "" &
                    participant.label != "",
                  .(session.code,
                    participant.label,
                    participant.time_started,
                    dPGG.1.player.contribution)]


meta_long[,
          `:=`(date = participant.time_started %>% ymd_hms() %>% date(),
               hour = participant.time_started %>% ymd_hms() %>% hour())]

meta <- meta_long[,
                  .(date = date %>% unique(),
                    time = hour %>% unique() %>% paste("00", sep = ":"),
                    showups = participant.time_started %>% length(),
                    dropouts = dPGG.1.player.contribution %>% is.na() %>% sum(na.rm = FALSE)),
                  by = session.code]

meta[,
     `:=`(participants = showups - dropouts,
          observations = (showups - dropouts)/4)]

# save data
fileName <- "replication2021"
save(meta, file = paste0("../../../data/processed/rda/", fileName, "_meta", ".rda"))
write.csv(meta, file = paste0("../../../data/processed/csv/", fileName, "_meta", ".csv"))

rm(list = c("full", "meta_long"))
```

```
sample_info <- read.table(file = "../../../data/sample/invited_students.txt",
                          col.names = c("participant.label")) %>%
  data.table()
```

We conducted a series a 4 sessions in between Thursday, July 01 to Friday, July 23 and collected 29 observations (from 116 participants) in total. This replication is registered in the AEA RCT Registry and the unique identifying number is: AEARCTR-0007902 (Berlemann, Roggenkamp, and Traub 2021).

```
meta %>% kable()
```

session.code

date

time

showups

dropouts

participants

observations

jyf8xd0s

2021-07-01

15:00

35

7

28

7

vggk2gh1

2021-07-03

13:00

20

8

12

3

8gi7c8xg

2021-07-09

13:00

21

9

12

3

d6jrsxnr

2021-07-23

14:00

75

11

64

16

Unfortunately, w needed to recruit students for the last session as the sample of non-students was exhausted. As a consequence, we 59 students participated in the last session.

First round

Eventually, we'll be interested in the participants' first round's behavior, as it indicates their willingness to cooperate before they interact with one another. As a consequence, 116 participants yield 116 observations that shall be stored in a data table called `replicationFirstRound`. This table carries information on a participant's group members' contributions `othersContribution`, the participant's `ownContribution` and a trust measure[1] obtained from the personality test at the end of the experiment.

[1] "Generally, others can be trusted". Likert scale ranging from 0 (do not agree) to 6 (agree).

```
# create data table
replicationFirstRound <- DT[,
                  .(participant.code,
                    treatment = "replication",
                    session.code,
                    groupID = paste(session.code, dPGG.1.group.id_in_subsession, sep = "_"),
                    othersContribution = dPGG.1.group.total_contribution - dPGG.1.player.contribution,
                    ownContribution = dPGG.1.player.contribution,
                    trust = Outro.1.player.PQ11,
                    comprehension = dPGG.10.player.comprehension)]

# save data
save(replicationFirstRound, file = paste0("../../../data/processed/rda/", fileName, "_R1", ".rda"))
write.csv(replicationFirstRound, file = paste0("../../../data/processed/csv/", fileName, "_R1", ".csv"))

if(knitr::is_html_output()){
  # display data
  replicationFirstRound %>%
    head(n = 12) %>%
    kbl() %>%
    # scroll_box(height = "200px") %>%
    kable_paper("hover",
                full_width = TRUE,
                fixed_thead = TRUE)

  # create download button
  replicationFirstRound %>%
        download_this(
          output_name = glue("{Sys.Date()}_Simulation_firstRound"),
          output_extension = ".csv", # CSV output
          button_label = "Download csv",
          button_type = "default"
        )
  }
```

The data is saved in two formats (`csv` and `rda`) in `../../../data/processed/`. If you have not downloaded the repository and access the html file, you can also obtain a csv-file by clicking on the button above (HTML-only feature).

All rounds

Because we are most interested in the dynamics of the game, the most important data frame shall reflect the interactions within groups over time. I'll subset the data using a regular expression such that one obtains the initial belief as well as endowments, contributions, gains and stocks for each individual (preliminarily).[2] [3]

```
# most relevant variables
mRegex <- "participant\\.code$|session\\.code$|dPGG\\.1\\.group\\.id_in_subsession|^dPGG.1.player.id_in_
mainVariables <- str_subset(string = names(DT),
                            pattern = mRegex)
subset <- DT[, ..mainVariables]
```

[2] There will also be information on the respondent's self stated comprehension of the task as well as a dummy indicating whether a bot was active (which implies that some group member dropped out.)

[3] In addition the data table contains a `participant.code`, a `session.code` as well as some group and individual ID for identification and merging purposes.

Because the the endowments are dynamic, the contributions may vary within and across groups. For this reason, a `share` (reporting the share of a respondent's current endowment contributed) is calculated. The first few rows of the data table look as follows:

```
# refactor groupID such that it also contains treatment-info
subset[,
   groupID := paste(session.code, dPGG.1.group.id_in_subsession,
                    sep = "_")]

# add share as contribution/endowment
for(round in 1:10){
  contribution <- glue("dPGG.{round}.player.contribution")
  endowment <- glue("dPGG.{round}.player.endowment")
  subset[, glue("dPGG.{round}.player.share") := subset[[contribution]]/subset[[endowment]] ]
}

# add treatment variable
subset[,
       treatment := "replication"]

if(knitr::is_html_output()){
  # display data
  subset %>%
    head(n = 12) %>%
    kbl() %>%
    scroll_box(width = "100%") %>%
    kable_paper("hover",
                full_width = TRUE,
                fixed_thead = TRUE)
  }
```

To obtain a long (instead of a wide) table that reports these variables for independent observations over time, the data hast be be transformed and aggregated. Instead of listing individuals, it shall list groups over time.[4]

This process takes a few steps:

```
cluster <- c("treatment", "session.code", "groupID")
outcomes <- c("contribution", "endowment", "share", "stock", "gain", "bot_active")
# outcome = "contribution"
DTs <- list()
for(outcome in outcomes){
  if(outcome == "bot_active"){
    var = names(subset) %>% str_subset(pattern = glue("group\\.{outcome}$"))
  } else {
    var = names(subset) %>% str_subset(pattern = glue("player\\.{outcome}$"))
  }


  # calculate either averages or the sum per round per group
  if(outcome == "share"){
    aggregates = subset[,
                lapply(.SD, mean, na.rm=TRUE),
                by = cluster,
                .SDcols=var
```

[4]These steps are eliminating lots of information. If one desires an extensive data table containing individual- and group-level information over time, one can merge the resulting table with the small grained data.

```
                ]
  } else {
    aggregates = subset[,
                lapply(.SD, sum, na.rm=TRUE),
                by = cluster,
                .SDcols=var
                ]
  }


  # transform from wide to long
  meltedAggregates <- melt(aggregates, id.vars = cluster, measure.vars = var)
  DTname <- glue("{str_to_title(outcome)}")
  DTs[[DTname]] <- meltedAggregates
  rm(list = c("DTname", "meltedAggregates", "aggregates", "var", "outcome"))
}
```

```
for(i in 1:length(outcomes)){
  DTs[[i]] <- DTs[[i]][,
                       .(treatment,
                         session.code,
                         groupID,
                         round = str_replace_all(string = variable,
                                                 pattern = "\\D",
                                                 replacement="") %>% as.integer(),
                         value # to be renamed afterwards
                       )
  ]
  # rename "value" to outcome variable
  setnames(DTs[[i]], old = "value", new = outcomes[i])
}
```

Also, the Gini coefficient has to be calculated to measure inequality within groups.

```
# note that GMTV used start of period earnnings, i.e. endowments. We use end of period earnings, i.e. s
# this adjustment has been considered in our processing of GMTVs data.

var = names(subset) %>% str_subset(pattern = "player\\.stock$")
gini = subset[,
              lapply(.SD, Gini, na.rm=TRUE),
                by = cluster,
                .SDcols=var
              ]
Gini <- melt(gini, id.vars = cluster, measure.vars = var)

DTs[["Gini"]] <- Gini[,
                      .(treatment,
                        session.code,
                        groupID,
                        round = str_replace_all(string = variable,
                                                pattern = "\\D",
                                                replacement="") %>% as.integer(),
                        gini = value
                         )]
rm(list = c("var", "gini", "Gini"))
```

Finally, the data that was stored in a list will be reduced to a single data table.

```
replication <- Reduce(function(...) merge(..., by=c(cluster, "round"), all = TRUE), DTs)
```

Next, we'll flag groups that are richer and poorer than the median group.

```
median <- replication[round == 10,
                      median(stock)]

richGroups <- replication[round == 10 & stock > median,
                          unique(groupID)]

poorGroups <- replication[round == 10 & stock < median,
                          unique(groupID)]

replication[groupID %in% richGroups,
            rich := TRUE]

replication[groupID %in% poorGroups,
            rich := FALSE]
```

```
# flag observations where at least one participant did not understand the game
noComp <- subset[dPGG.10.player.comprehension == 0,
                 groupID] %>% unique()
replication[,
     noComprehension := 0]
replication[groupID %in% noComp,
     noComprehension := 1]


# drop observations (i.e. groups in rounds) with dropouts (bot_active == 1) and
# where round > 10
replication <- replication[bot_active == 0 & round <= 10]
```

Unfortunately, I made a mistake calculating the `share`. For this reason, it will be recalculated here:

```
replication[, share := contribution/endowment]
```

The result contains `NROW(DT)/4` observations in 10 rounds and is saved as a `csv` and as a `rda` file in `../../../data/processed/`.

```
save(replication, file = paste0("../../../data/processed/rda/", fileName, ".rda"))
write.csv(replication, file = paste0("../../../data/processed/csv/", fileName, ".csv"))
```

The first few rows look as follows. The full data set can be downloaded with a click on the button below (in case you only have access to this `html` file).

```
if(knitr::is_html_output()){
  replication %>%
    head(n = 12) %>%
    kbl() %>%
    # scroll_box(height = "200px") %>%
    kable_paper("hover",
                full_width = TRUE,
                fixed_thead = TRUE)
  }
```

```r
if(knitr::is_html_output()){
  replication %>%
        download_this(
          output_name = glue("{Sys.Date()}_Simulation"),
          output_extension = ".csv", # CSV output
          button_label = "Download csv",
          button_type = "default"
        )
  }
```

Covariates

```r
# add variables
DT[, treatment := "replication"]

DT[, groupID := paste(session.code, dPGG.1.group.id_in_subsession,
                      sep = "_")]

# subset
cRegex <- "participant.code|session.code|treatment|groupID|Outro.1.player|10.player.donation|switching_
covariates <- str_subset(string = names(DT),
                              pattern = cRegex)
CT <- DT[, ..covariates]

# rename
names(CT) <- names(CT) %>%
  str_replace_all(pattern =".*player\\.",
                  replacement = "") %>%
  str_to_lower()

names(CT)[names(CT) == "groupid"] <- "groupID"

# refactor
CT[, donation := donation/20]
CT[donation %>% is.na, donation := 0]

CT[, gender := ifelse(test = gender == "female",
                      yes  = 1,
                      no   = 0)]

CT[, inconsistent := as.logical(inconsistent)]

replicationCovariates <- CT

# write data
fileName <- "replication2021"
save(replicationCovariates,
     file = paste0("../../../data/processed/rda/", fileName, "_COVS", ".rda"))
write.csv(replicationCovariates,
          file = paste0("../../../data/processed/csv/", fileName, "_COVS", ".csv"))

# no display or download button as these data may qualify as PII
```

Page Times

```
files <- list.files(path = "../../../data/pageTimes/",
                        recursive = TRUE,
                        full.names = TRUE)
csvs <- list()

for(i in files){
  name <- str_extract(string = i,
                      pattern = "2021.*")

  temp <- read.csv(i,
                   stringsAsFactors = FALSE) %>%
    data.table()

  csvs[[name]] <- temp
}

timeSpent <- rbindlist(l = csvs,
                       use.names = TRUE) %>%
  data.table()

setorder(timeSpent, session_code, participant_code, epoch_time)
```

```
timeSpent[,
          lag := shift(epoch_time, fill = NA, type = "lag"),
          by = c("session_code", "participant_code")]

timeSpent[,
          duration := epoch_time - lag,
          by = c("session_code", "participant_code")]

timeSpent[,
          completion := epoch_time %>% max() - epoch_time %>% min(),
          by = c("session_code", "participant_code")]

duration <- timeSpent[participant_code %in% DT$participant.code,
                      .(
                        session_code,
                        participant_code,
                        app_name,
                        page_name,
                        page_index,
                        page_submission = epoch_time,
                        time_spent = duration,
                        completion_time = completion
                      )]

save(duration, file = paste0("../../../data/processed/rda/", fileName, "_timeSpent", ".rda"))
write.csv(duration, file = paste0("../../../data/processed/csv/", fileName, "_timeSpent", ".csv"))
```

Outlook

The next step is to reproduce Gaechter et al.'s figures and tables and to compare our results to their results. This will be done in the third report.

Berlemann, Michael, Hauke Roggenkamp, and Stefan Traub. 2021. "Replication: Growth and Inequality in Public Good Provision (No-Punish-10) by Gächter Et Al. (2017)." *AEA RCT Registry.* https://doi.org/https://doi.org/10.1257/rct.7902-2.0.

Chen, Daniel L., Martin Schonger, and Chris Wickens. 2016. "oTree-an Open-Source Platform for Laboratory, Online, and Field Experiments." *Journal of Behavioral and Experimental Finance* 9: 88–97. https://doi.org/10.1016/j.jbef.2015.12.001.

Gächter, Simon, Friederike Mengel, Elias Tsakas, and Alexander Vostroknutov. 2017. "Growth and Inequality in Public Good Provision." *Journal of Public Economics* 150: 1–13. https://doi.org/10.1016/j.jpubeco.2017.03.002.