Merge GUK files

April 10, 2017
19:46

Seiro Ito

Contents

Based on data received in March, 2017.

Attrition Attrition rate who disappeared by rd 3 is 5.8%. Judging from HH head names, majority of visited households seem to be the same households. Respondent names change that cannot be explained by typo's.

Treatment assignment For surviving subjects, treatment assignment given in treatment in loan_status.dta matches with the original assignment given in treat of treatment assignment instruction file. By rd 2, majority of the control started to receive treatments. It took about 250 days to complete the treatment on the treated, only then the treatment on the control started. There are 29 individuals that attrited due to individual rejection, 26 individuals attrited due to group rejection. Some of the individuals rejected to be treated stay in the sample.

Arm Little discrepancy in arm consistency in the identification file.

Food consumption Missing observations match with attrition. No problem found.

I produced following files (all tab-separated text files) after various correctins described in this file:

attrition.prn An exhaustive list of households and their observation status file.

interview_dates_long.prn, interview_dates_wide.prn Interview date files in long and wide formats.

idfu.prn Cover page information files.

treatment_assignment.prn Treatment assignment information using all the above three files and loan status file ./1/original/loan_status_october15-2015.prn.

I read

List folders and files.

```
setwd(pathsource.mar)
foldername ← list.dirs(path = ”.”, recursive = T, full.names = T)
foldername ← foldername[!grepl(skip.in.rd1, foldername)]
fn ← list.files(path = foldername, pattern = ”.prn$”,
    recursive = T, full.names = T)
fn ← unique(fn)
#There will be warnings due to duplicated factor levels in dta which should be a fair warr
```

```
setwd(pathsource.mar)
Z ← lapply(fn, fread, integer64 = ”double”)
```

```
fnids ← grepout(”\\/id”, fn) # grep(a,b) returns matched string of a in b
fnids ← fnids[!grepl(”rev”, fnids)]
```

Read individual identification files in ./1/original/identification_p1_2012-2013.prn, ./2/identification_p2.prn, ./3/identification_p3.prn.

```
setwd(pathsource.mar)
fn ← list.files(path = foldername, pattern = ”.prn$”,
    recursive = T, full.names = T)
fn ← unique(fn)
fnids ← grepout(”\\/id”, fn)
fnids ← fnids[!grepl(”rev”, fnids)]
idfiles ← lapply(1:3, function(i) cbind(rd = i, Z[fn %in% fnids][[i]]))
```

Column names with hh, rd, gid, name.

```
lapply(idfiles, function(x) grepout(”hh|rd|gid|name”, colnames(x)))
```

```
[[1]]
[1] "rd"       "hhid"     "gid"      "rname"    "hhh_name" "g_name"   "e_name"
[8] "tl_name"

[[2]]
[1] "rd"       "hhid"     "gid"      "r_name"   "hhh_name" "hh_id"    "e_name"
[8] "s_name"   "deo_name"

[[3]]
[1] "rd"       "hhid"     "gid"      "r_name"   "hhh_name" "hh_id"    "e_name"
[8] "s_name"   "deo_name"
```

Rds with hh_id, variable name corrected to hhid.

```
c(unlist(lapply(idfiles, function(x) any("hh_id" %in% colnames(x)))),
unlist(lapply(idfiles, function(x) any("hhid" %in% colnames(x)))))
```

```
[1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
#Rds without \textsf{hhid} (named as \textsf{id}).
setnames(idfiles[[1]], c("rname", "tl_name", "zila", "cell"),
        c("r_name", "s_name", "zilla", "cell_no"))

lapply(idfiles, function(x)
        setnames(x, grepout("\\_", colnames(x)), gsub("\\_", ".", grepout("\\_", colnames(
```

Households are added after immediately after rd 1 original sample. They have hhid headed with 98 or 99.

II merge identification files (cover page information)

Found: Rate of attrition who disappeared by rd 3 is 5.8%. Judging from HH head names, majority of visited households seem to be the same households. Respondent names change that cannot be explained by typo's.

Bind the files to get a long formated panel data.

```
idf ← do.call("rbind", c(idfiles, fill = T))
setkey(idf, hhid, rd)
setnames(idf, c("r.name", "e.name", "s.name"), c("rname", "ename", "sname"))
```

II.1 hhid duplication correction

Number of duplicated entries.

```
table0(!duplicated(idf[, .(rd, hhid)]))
```

```

FALSE  TRUE
    4  6386
```

These 4 duplicated entries are:

```
idf[hhid %in% hhid[duplicated(idf[, .(rd, hhid)])],
        .(rd, hhid, gid, village, rname, hhh.name)]
```

```
    rd        hhid    gid         village    rname hhh.name
 1:  1     7137302  71373       roton pur  sonavan     kalu
 2:  2     7137302  71373  uttar ratonpur   goleja  mofidul
 3:  2     7137302  71373        rotonpur  sonavan kalu mia
 4:  3     7137302  71373 north roton pur sonavanu kalu mia
 5:  1     7137316  71373       roton pur   shathi   shathi
 6:  2     7137316  71373        rotonpur    shati    shati
 7:  3     7137316  71373 north roton pur  monoara    karim
 8:  3     7137316  71373        rotonpur    sathi    sathi
 9:  1     7137317  71373       roton pur  roshida    rafiq
10:  2     7137317  71373        rotonpur   rosida    rofiq
11:  3     7137317  71373       roton pur   rosida    rofiq
12:  3     7137317  71373 north roton pur   afruza    oahed
13:  1 99081912418 819124       bill para   zoneka     anis
14:  2 99081912418 819124       bill para    rezia    rezia
15:  2 99081912418 819124       bill para   zoneka     asis
16:  3 99081912418 819124        billpara   zoneka     anis
```

Duplication is found in hhid.

- rname of 99081912418 in rd 2 is zoneka, rezia is 99081912406 whose all rds exist, so drop 99081912418, rezia.
- 7137302, ”goleja” is 7137305.
- 7137316, ”monoara” is not found in other rounds. Change hhid to 9997137316.
- 7137317, ”afruza” is not found in other rounds. Change hhid to 9997137317.

```
idf[hhid == 7137302 & grepl(”gole”, rname), hhid := 7137305]
idf ← idf[!(hhid == 99081912418 & grepl(”rezi”, rname)), ]
idf[hhid == 7137316 & grepl(”monoa”, rname), hhid := 9997137316]
idf[hhid == 7137317 & grepl(”afruz”, rname), hhid := 9997137317]
```

Confirm duplication is gone.

```
table0(!duplicated(idf[, .(rd, hhid)]))
```

```
TRUE
6389
```

```
setkey(idf, hhid, rd)
idf[, c(”hh.id”, ”village.code”, ”union.code”, ”zilla.code”) := NULL]
```

II.2 hhid typo correction

Further correct hhid. Changed in hhid:

- 990817 → 990819
- 98.*544 → 98.*545
- Drop 980: From 9808159108, 9808159119, 9808159120, 9808159202, 9808159212 → 8159108, 8159119, 8159120, 8159202, 8159212
- 7044308 → 7044608
- 98070511 → 99070511
- 9807*→9907*:
  - In village names with koachkhali, koachakhali, koach khali, koachkhaly, west koachkhali, east koskhali, east koachkhali, east koach khali, east koachkhaali, koachkhali west, east koachkhaly (kosk|koach) and hhid with 9807, or 9807031602, 9807031606, 9807031612, 9807031615, 9807031620.
  - (Corrected in March cleaning) In village names with uttor vatiapara, north vatiapara, uttor vatiya para, uttor vatiyapara, uttor vatia para (^u.*vati|^n.*vati), 99070210901, 99070210902, 99070210903, 99070210904, 99070210905, 99070210906, 99070210907, 99070210908, 99070210909, 99070210910, 99070210911, 99070210912, 99070210913, 99070210914, 99070210915, 99070210916, 99070210917, 99070210918, 99070210919, 99070210920.
  - (Corrected in March cleaning) In village names with hasdhora, hasddhora (hasd) and hhid with 71010, or 99071010801, 99071010802, 99071010803, 99071010804, 99071010805, 99071010806, 99071010807, 99071010808, 99071010809, 99071010810, 99071010811, 99071010812, 99071010813, 99071010814, 99071010815, 99071010816, 99071010817, 99071010818, 99071010819, 99071010820.
- (Corrected in March cleaning) 311→911: In village names with harudagga, harudanga (harud) and hhid with 60 or 61, or 99070911601, 99070911602, 99070911603, 99070911604, 99070911605, 99070911606, 99070911607, 99070911608, 99070911609, 99070911610, 99070911611, 99070911612, 99070911613, 99070911614, 99070911615, 99070911616, 99070911617, 99070911618, 99070911619.
- (Corrected in March cleaning) 817→819: In village names with bill para, billpara (bill) and hhid with 240 or 241, or 99081912401, 99081912402, 99081912403, 99081912404, 99081912405, 99081912406, 99081912407, 99081912408, 99081912409, 99081912410, 99081912411, 99081912412, 99081912413, 99081912414, 99081912415, 99081912416, 99081912417, 99081912418, 99081912419.

```
idf[, hhid := asc(hhid)]
idf[grepl(990817, hhid), hhid := gsub(”990817”, ”990819”, hhid)]
idf[grepl(”98.∗544”, hhid), hhid := gsub(”544”, ”545”, hhid)]
idf[grepl(”98070511”, hhid), hhid := gsub(”98070511”, ”99070511”, hhid)]
idf[grepl(”9808159108|9808159119|9808159120”, hhid),
        hhid := gsub(”9808159”, ”8159”, hhid)]
idf[grepl(”9808159202|9808159212”, hhid),
        hhid := gsub(”98081592”, ”81591”, hhid)]
idf[grepl(”^u.∗vati|^n.∗vati”, village) & grepl(”^98”, hhid),
        hhid := gsub(”9807”, ”9907”, hhid)]
idf[grepl(”kosk|koach”, village) & grepl(3115, hhid),
        hhid := gsub(”9807”, ”9907”, hhid)]
idf[grepl(”hasd”, village) & grepl(71010, hhid),
        hhid := gsub(”9807”, ”9907”, hhid)]
idf[grepl(”harud”, village) & grepl(”60|61”, hhid),
        hhid := gsub(”311”, ”911”, hhid)]
idf[grepl(”bill”, village) & grepl(”240|241”, hhid),
        hhid := gsub(”817”, ”819”, hhid)]
idf[, hhid := asn(hhid)]
```

Check how many hhids constitutute complete panel. Tabulate number of repeated observations in a panel (1: observed only once, 2: twice, 3: complete panel).

```
table(table0(idf[, hhid]))
```

```

   1    2    3 
 102   70 2049 
```

II.3 create some variables for convenience

Create memstatus for covenience and understandability.

```
idf[, memstatus := tolower(membership.status)]
idf[grepl(”epl”, memstatus), memstatus := ”replacement”]
idf[grepl(”new”, memstatus), memstatus := ”new group”]
idf[grepl(”old”, memstatus), memstatus := ”old”]
idf[grepl(”ero”, memstatus), memstatus := ”lost to flood”]
idf[, memstatus := factor(memstatus, levels =
        c(”old”, ”replacement”, ”new group”, ”individual rejection”,
        ”group rejection”, ”lost to flood”))]
```

Here is how membership.status and memstatus correspond:

```
table0(idf[, .(membership.status, memstatus)])
```

```
                        memstatus
membership.status         old replacement new group individual rejection
  Group Erosion             0           0         0                    0
  Group Rejection           0           0         0                    0
  Individual Rejection      0           0         0                  413
  Individual Replacement    0         466         0                    0
  New Group                 0           0      1364                    0
  Old Member             3585           0         0                    0
                        memstatus
membership.status        group rejection lost to flood
  Group Erosion                        0           189
  Group Rejection                    372             0
  Individual Rejection                  0          0
  Individual Replacement                0          0
  New Group                             0          0
  Old Member                            0          0
```

Create arm for covenience and understandability.

```
idf[, arm := tolower(rand.arm)]
idf[grepl(”^L.*t$”, rand.arm), arm := ”large”]
idf[grepl(”^L.*gr”, rand.arm), arm := ”large grace”]
idf[grepl(”^p”, rand.arm), arm := ”cow”]
idf[grepl(”tr”, rand.arm), arm := ”traditional”]
# below are not strictly arms...
idf[grepl(”floo”, memstatus), arm := ”lost to flood”]
idf[rd == 1, arm := ”before intervention”]
table0(idf[, arm])
```

```

                     before intervention                  cow                large
                 232                2219                  951                  909
         large grace       lost to flood          traditional
                 928                 109                 1041
```

Subjects with no arm information is group rejection. Based on Char Randomization_2012.xlsx, arms are large grace, large, large, traditional, traditional, traditional, traditional, respectively for gids 70314, 70317, 70319, 70539, 70858, 81483, 81697.

```
table0(idfu[grepl(”gr.*rej”, memstatus), gid])
```

```

70314 70317 70319 70539 70858 81483 81697
   20    59    60    55    60    60    58
```

```
(rejgid ← unique(idf[grepl(”gr.*rej”, memstatus), gid]))
```

```
[1] 70314 70317 70319 70539 70858 81483 81697
```

```
rejarm ← c(”large grace”, ”large”, ”large”, rep(”traditional”, 4))
for (i in 1:length(rejgid)) idf[gid == rejgid[i], arm := rejarm[i]]
# If I run this (impose before intervention for all rd 1 obs)
# idf[rd == 1, arm := "before intervention"]
# then large grace will be overwritten for 70314.
# Reason: There are 2 groups under 70314, one which rejected and
# another which was formed after it.
idf[, arm := factor(arm)]
```

Group rejecters' arms:

```
table0(idf[grepl(”gr.*rej”, memstatus), arm])
```

```

      large large grace traditional
        119          20         233
```

Note that arm is defined using the information at the time of interview. So in rd 1, everyone is "before intervention."

II.4 missingness pattern across rounds

Check the missingness patterns. I will define the first observaton of additional household which were added to supplement households whose chars were eroded as rd 1. The timing of their visit was later than the first round of original households. Create an exhaustive hhid list, iu, and check existentce in each rd from idf. Create an attrition detector exist.

```
idunion ← unique(idf[, hhid])
idunion ← idunion[order(idunion)]
for (i in 1:3) assign(paste0(”i”, i), idunion %in% idfiles[[i]][ ,hhid])
iu ← data.table(idunion, i1, i2, i3)
iu[, exist := ””]
iu[i1, exist := 1]
iu[i2, exist := paste0(exist, 2)]
iu[i3, exist := paste0(exist, 3)]
iu[, exist := factor(exist, levels = c(1, 12, 13, 123))]
setnames(iu, ”idunion”, ”hhid”)
```

Save hhid list.

```
setwd(pathsave)
write.tablev(iu, ”attrition.prn”)
iu[, c(”i1”, ”i2”, ”i3”) := NULL]
```

Merge missingness pattern iu to idf.

```
setkey(iu, hhid); setkey(idf, hhid, rd)
idfu ← iu[idf]
```

Create last observed rounds of each hhid.

```
idfu[, last.rd := asn(substr(exist, nchar(asc(exist)), nchar(asc(exist))))]
table(idfu[rd == last.rd, exist])
```

```

   1   12   13  123
  99   29   42 2029
```

There are 99 individuals who were last observed in rd 1. There are also 29 who were also last seen in rd 2 who may also be drop outs. All other 2071 are observed in rd 3 (Note: 2071 are exist = 13, 23, 123, so it is different from counts of complete panel in iu). If we consider these drop outs as attrition, the attrition rate up to rd 3 is 5.82%. There are 128 individuals dropped out eventually.

II.5 arms

Last observed rd and arms. If last.rd == 1, arm is before intervention, large grace, traditional.

```
(tb.last ← table0(idfu[rd == last.rd, .(arm, last.rd)]))
```

```
                           last.rd
arm                          1   2   3
  before intervention       76   0   0
  cow                        0  12 478
  large                      0   4 476
  large grace               21   5 467
  lost to flood              0   2  54
  traditional                2   6 596
```

Check these 2 obs in traditional who dropped out in rd 1.

```
idfu[last.rd == 1 & grepl(”tra”, arm),
        .(gid, village, exist, rd, hhid, rname, arm, rand.arm, memstatus, st.day)]
```

```
     gid       village exist rd    hhid   rname         arm rand.arm
1: 70539 dakkin golna     1  1 7053909 shefali traditional       NA
2: 81697    naya char     1  1 8169719  rasida traditional       NA
         memstatus st.day
1: group rejection     NA
2: group rejection     NA
```

In rd 1, all subjects are given arms “before intervention.” Correct these 2 and confirm correction.

```
idfu[last.rd == 1 & grepl(”tra”, arm), arm := ”before intervention”]
(tb.last ← table0(idfu[rd == last.rd, .(arm, last.rd)]))
```

```
                      last.rd
arm                      1   2   3
  before intervention   78   0   0
  cow                    0  12 478
  large                  0   4 476
  large grace           21   5 467
  lost to flood          0   2  54
  traditional            0   6 596
```

From rd 2 onwards, membership.status gives old member, replaced member, individually rejected member, rejected group’s member, lost to flood, and new group. In rd1, eventual membership status is given (confirm this by tabulating at rd 1).

```
table0(idfu[rd == 1, membership.status])
```

```

         Group Erosion        Group Rejection   Individual Rejection
                    80                    140                    159
Individual Replacement              New Group             Old Member
                   159                    460                   1221
```

Check if exist (missingness pattern) is consistent with memstatus. Most of exist == 1 are rejecters and flood victims, other are attritions (“old”).

```
table0(idfu[rd == last.rd, .(memstatus, exist)])
```

```
                      exist
memstatus                 1   12   13  123
  old                    25   13   18 1165
  replacement             2    1    6  150
  new group               4    2    4  430
  individual rejection   22    7   13  117
  group rejection        22    4    0  114
  lost to flood          24    2    1   53
```

Check if treatment is consistent across rounds.

```
table0(idfu[, .(rd, arm)])
```

```
   arm
rd  before intervention  cow large large grace lost to flood traditional
  1                2061    0    40          40             0          78
  2                   0  471   493         461            55         598
  3                   0  480   495         467            54         596
```

```
#table0(idfu[rd == last.rd, .(memstatus, arm)])
```

There are some observations which has arm information in rd 1. They are group rejecters, and a new group in 70314 which was wrongly given the same gid as the old group in the same village.

```
table(idfu[rd == 1 & !grepl(”bef”, arm), memstatus])
```

```
                 old          replacement             new group
                   0                    0                    20
individual rejection      group rejection         lost to flood
                   0                  138                     0
```

Note that there are 40 observations (equal to 2 groups) for 70314.

```
table(idfu[rd == 1 & !grepl(”bef”, arm) & grepl(”gro”, memstatus), .(gid, arm)])
```

```
       arm
gid     before intervention cow large large grace lost to flood traditional
  70314                   0   0     0          40             0           0
  70317                   0   0    20           0             0           0
  70319                   0   0    20           0             0           0
  70539                   0   0     0           0             0          19
  70858                   0   0     0           0             0          20
  81483                   0   0     0           0             0          20
  81697                   0   0     0           0             0          19
```

```
table(idfu[rd == 1 & gid == 70314, .(memstatus, arm)])
```

```
                      arm
memstatus              before intervention cow large large grace lost to flood
  old                                    0   0     0           0             0
  replacement                            0   0     0           0             0
  new group                              0   0     0          20             0
  individual rejection                   0   0     0           0             0
  group rejection                        0   0     0          20             0
  lost to flood                          0   0     0           0             0
                      arm
memstatus              traditional
  old                            0
  replacement                    0
  new group                      0
  individual rejection           0
  group rejection                0
  lost to flood                  0
```

Correct arm for 2nd group (memstatu==new) with gid 70314 (the one formed after the predecessor rejected the treatment) in rd 1 as “before intervention”.

```
idfu[rd == 1 & gid == 70314 & grepl(”new”, memstatus), arm := ”before intervention”]
```

Tabulation after correction:

```
table0(idfu[!grepl(”^gro”, memstatus), .(rd, arm)])
```

```
   arm
rd  before intervention  cow large large grace lost to flood traditional
  1                2079    0     0           0             0           0
  2                   0  471   453         461            55         520
  3                   0  480   456         467            54         521
```

Cover page has arm assignment under rand_arm. I derived arm from it for convenience. First, check if there is a change in arms. To do so, I will check, among non group rejecters which I omit from the consideration, if there is more than one arm recorded for nonNA lines for a given hhid. Since rd 1 is before intervention, there should not be more than 2 types of arm.

```
idfu[, arm.number := length(unique(arm)), by = ”hhid”]
table0(idfu[!grepl(”^gro”, memstatus) & arm.number < 2, rd])
```

```

 1  3
78  2
```

These are the observations found only in one rd. Most are observed in rd 1, while other 2 are in rd 3. These 2 obs are arm wth traditional that we saw earlier in this subsection.

```
idfu[!grepl(”^gro”, memstatus) & arm.number < 2 & rd > 1,
        .(gid, village, hhid, rd, exist, rname, arm, memstatus, d.1st)]
```

```
     gid         village       hhid rd exist   rname arm memstatus  d.1st
1: 71373 north roton pur 9997137316  3    NA monoara cow       old  11115
2: 71373 north roton pur 9997137317  3    NA  afruza cow       old 301115
```

Monoara in rd 3 may be hhid 713720, not 9997137316.

```
idfu[gid == 71373 & grepl(”ara|min”, rname),
        .(gid, village, hhid, rd, exist, rname, arm, memstatus, d.1st)]
```

```
     gid         village       hhid rd exist    rname                 arm
1: 71373       roton pur    7137313  1   123  anoyara before intervention
2: 71373  uttar rotonpur    7137313  2   123  anowara                 cow
3: 71373 north roton pur    7137313  3   123   jesmin                 cow
4: 71373       roton pur    7137320  1     1 monoyara before intervention
5: 71373 north roton pur 9997137316  3    NA  monoara                 cow
   memstatus  d.1st
1:       old     NA
2:       old  11014
3:       old 291115
4:       old     NA
5:       old  11115
```

Change 9997137316 in rd3 to 713720?

II.6 membership status

Check memstatus changes.

```
idfu[, memstatus.change := F]
idfu[, memstatus.change := length(unique(memstatus)) > 2, by = ”hhid”]
table0(idfu[, memstatus.change])
```

```
FALSE
 6389
```

II.7 rejections

Some subjects rejected to be treated. Their memberships were replaced with newly recruited members. One sees that individual rejection is replaced with replacement in each group. (And there is at least one rejecter in a group...)

```
idfu[, rejInGroup := any(grepl(”in.*rej”, memstatus)), by = gid]
idfu[, rejOfGroup := any(grepl(”g.*rej”, memstatus)), by = gid]
table0(idfu[rd == last.rd & rejInGroup, .(memstatus, gid)])
```

```
                      gid
memstatus              70204 70209 70315 70316 70318 70420 70421 70422 70425
  old                     17    19    15    14    13    17    19    19    16
  replacement              3     1     5     6     7     3     1     1     4
  new group                0     0     0     0     0     0     0     0     0
  individual rejection     3     1     5     6     7     3     1     1     4
  group rejection          0     0     0     0     0     0     0     0     0
  lost to flood            0     0     0     0     0     0     0     0     0
                      gid
memstatus              70427 70436 70541 70543 70545 70646 70650 70652 70653
  old                      9    19    19    13    16     9    11     3     9
  replacement             11     1     1     7     4    11     9    17    11
  new group                0     0     0     0     0     0     0     0     0
  individual rejection    11     1     1     7     4    11     9    17    11
  group rejection          0     0     0     0     0     0     0     0     0
  lost to flood            0     0     0     0     0     0     0     0     0
                      gid
memstatus              70757 70859 70861 71065 71268 71271 71335 71372 81478
  old                     19    17    18    17    18    16    13    14    16
  replacement              1     3     2     3     2     4     7     6     4
  new group                0     0     0     0     0     0     0     0     0
  individual rejection     1     3     2     3     2     4     7     6     4
  group rejection          0     0     0     0     0     0     0     0     0
  lost to flood            0     0     0     0     0     0     0     0     0
                      gid
memstatus              81479 81480 81482 81588 81591 81696 81698 817101 817103
  old                     19    10    18    18    19    19    19     19     16
  replacement              1    10     2     2     1     1     1      1      4
  new group                0     0     0     0     0     0     0      0      0
  individual rejection     1    10     2     2     1     1     1      1      4
  group rejection          0     0     0     0     0     0     0      0      0
  lost to flood            0     0     0     0     0     0     0      0      0
                      gid
memstatus              817105
  old                      19
  replacement               1
  new group                 0
  individual rejection      1
  group rejection           0
  lost to flood             0
```

Check # of rejection = # of replacement.

```
setkey(idfu, gid)
idfu[rd == last.rd, numIndRej := sum(grepl(”ind.*rej”, memstatus)), by = gid]
idfu[rd == last.rd, numRepl := sum(grepl(”repla”, memstatus)), by = gid]
nrow(idfu[rd == last.rd & numIndRej != numRepl, ])
```

```
[1] 0
```

Group rejection is given in the below.

```
table0(idfu[rd == last.rd & rejOfGroup, .(arm, gid)])
```

```
                     gid
arm                   70314 70317 70319 70539 70858 81483 81697
  before intervention     1     0     0     1     0     0     1
  cow                   0     0     0     0     0     0     0
  large                 0    20    20     0     0     0     0
  large grace          39     0     0     0     0     0     0
  lost to flood         0     0     0     0     0     0     0
  traditional           0     0     0    19    20    20    19
```

gid 70314 has 40 entries. This is group rejection and new group under the same gid, which should not... As we saw earlier when we corrected arm of 30714 in rd 1 to “before intervention.”

```
table(idfu[gid == 70314 & rejOfGroup & grepl(”bef”, arm), .(exist, memstatus)])
```

```
     memstatus
exist old replacement new group individual rejection group rejection
  1     0           0         1                    0               0
  12    0           0         0                    0               0
  13    0           0         0                    0               0
  123   0           0        19                    0               0
     memstatus
exist lost to flood
  1               0
  12              0
  13              0
  123             0
```

According to JDS paper (p.13), group rejection happened only after groups learned their arms.

Save idfu.

```
setwd(pathsave)
write.tablev(idfu, ”idfu.prn”)
saveRDS(idfu, ”idfu.rds”)
```

idfu: idf (cover page) + iu (an exhaustive hhid list).

II.8 validate if names stay the same in the same hhid (suggestive but not defenite to correct hhid)

Reshape to see if names differ. I assign 123 if names are same in all rounds, for respondent names and hh head names.

```
iic ← paste0(c(”hhid”, ”exist”, ”gid”, ”village”, ”last.rd”, ”po.name”, ”g.name”), collap
idfu2 ← idfu[, grepout(”hh|rd|village$|gid|status|name|exist|memst|type|G|arm”,
        colnames(idfu)), with = F]
idchk ← reshape(idfu2, direction = ”wide”,
        idvar = c(”hhid”, ”gid”, ”exist”, ”last.rd”),
        timevar = ”rd”, v.names = colnames(idfu2)[!grepl(paste0(iic, ”|rd”), colnames(idfu
idchk[, rname.chk := 0]
idchk[grepl(12, exist) &
        !is.na(rname.1) & !is.na(rname.2) & rname.1 == rname.2,
        rname.chk := 12]
idchk[grepl(23, exist) &
        !is.na(rname.2) & !is.na(rname.3) & rname.2 == rname.3,
        rname.chk := 23]
idchk[grepl(123, exist) &
        !is.na(rname.1) & !is.na(rname.2) & !is.na(rname.3) &
        rname.1 == rname.2 & rname.2 == rname.3,
        rname.chk := 123]
idchk[grepl(13, exist) &
        !is.na(rname.1) & !is.na(rname.3) &
        rname.1 == rname.3,
        rname.chk := 13]
idchk[, hname.chk := 0]
idchk[grepl(12, exist) &
        !is.na(hhh.name.1) & !is.na(hhh.name.2) & hhh.name.1 == hhh.name.2,
        hname.chk := 12]
idchk[grepl(23, exist) &
        !is.na(hhh.name.2) & !is.na(hhh.name.3) & hhh.name.2 == hhh.name.3,
        hname.chk := 23]
idchk[grepl(123, exist) &
        !is.na(hhh.name.1) & !is.na(hhh.name.2) & !is.na(hhh.name.3) &
        hhh.name.1 == hhh.name.2 & hhh.name.2 == hhh.name.3,
        hname.chk := 123]
idchk[grepl(13, exist) &
        !is.na(hhh.name.1) & !is.na(hhh.name.3) &
        hhh.name.1 == hhh.name.3,
        hname.chk := 13]
```

Respondent name consistency across rounds. 0 indicates no same name at all in all 3 rds. The following table shows tabulation of number non-matching cases across 3 rds against their missingness patterns. So there are 49 cases (hhid's) that have all respondent names different across rounds. Some of these cases are just typo's, but we need to confirm if this is the case. rname.1 shows rname in rd 1, rname.2 shows rname in rd 2, and so on. The respondent names are pretty different between rd 1 and 2,3. Rds 2-3 show similar names.

```
table0(idchk[, .(rname.chk, exist)])
```

```
         exist
rname.chk   1  12  13 123 <NA>
      0    99  17  26 281    1
      12    0  12   0 152    0
      13    0   0  16   0    0
      23    0   0   0 662    0
      123   0   0   0 934    0
```

```
idchk[rname.chk == 0 & exist == 123,
        grepout("rname\\.\\d", colnames(idchk)), with = F]
```

```
        rname.1  rname.2    rname.3
  1:     farida   farija     farida
  2:    morgina morigina    morgina
  3:  abu taleb abutaleb chondrobanu
  4:      yusuf   fojila     fozila
  5:   mohammod   sahera    mohamod
 ---
277:    shahina   sahina      sahia
278: abu siddik   raseda    rasheda
279:     zoynob  akkabor     joynob
280:   dhanmoty dhonmoty   dhanmoty
281:       asma   hasmot       asma
```

Length of rname.chk should be no shorter than exist if the respondent name matches in all rounds. There are 1138 cases. The cases that two match are 1061.

```
table0(idchk[, .(rname.chk, exist)])
```

```
         exist
rname.chk   1  12  13 123 <NA>
      0    99  17  26 281    1
      12    0  12    0 152     0
      13    0   0  16   0     0
      23    0   0   0 662     0
      123   0   0   0 934     0
```

HH head name consistency across rounds. 0 indicates no match at all.

```
table0(idchk[, .(hname.chk, exist)])
```

```
         exist
hname.chk    1   12   13  123 <NA>
      0     99   10   31  280    1
      12     0   19    0  205    0
      13     0    0   11    0    0
      23     0    0    0  391    0
      123    0    0    0 1153    0
```

In the following output, the first row is hhid, followed by reported head names in rds 1, 2, 3, 4.

```
print(idchk[hname.chk == 0  & exist == 123,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
            hhid hhh.name.1        hhh.name.2       hhh.name.3
  1:     7010111   nur alom           nurglam         nur alom
  2:     7020202  abu taleb          abutaleb        abu taleb
  3:  9807020405      babul             bablu            babul
  4:  9807020417     ajijul      ajijur haque           ajijur
  5:     7020615  abu talab          abutaleb        abu talab
 ---
276: 99081912120 abu siddik abu bokkor siddik abubokkor siddik
277: 99081912405       hira           hiralal         hira lal
278: 99081912417    taramia          tara mia          taramia
279: 99081912418       anis              asis             anis
280: 99081912419     hasmot              asma           hasmot
```

```
print(idchk[hname.chk == 0  & exist == 23,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
Empty data.table (0 rows) of 4 cols: hhid,hhh.name.1,hhh.name.2,hhh.name.3
```

```
NULL
```

```
print(idchk[hname.chk == 0  & exist == 13,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
           hhid hhh.name.1 hhh.name.2 hhh.name.3
 1:     7020312     ajahar         NA     azahar
 2:     7020412     shofik         NA       sofi
 3:     7021320       alim         NA      bablu
 4:     7042120       jalo         NA      .jalo
 5:  9807042103     shalam         NA      salam
---
27:    81710203    sukiron         NA   sukhiron
28: 98081710316 ero mondol         NA pirumondol
29: 98081710317  kala chan         NA hojrot ali
30:    81710504      zamaz         NA      zamal
31:    81710517    mogiber         NA    mojibor
```

```
print(idchk[hname.chk == 12  & exist == 12,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
           hhid hhh.name.1 hhh.name.2 hhh.name.3
 1:     7021220    mostofa    mostofa         NA
 2:     7042007      amzad      amzad         NA
 3:     7042515      anser      anser         NA
 4:     7043618 choimuddin choimuddin         NA
 5:     7053903      sobia      sobia         NA
---
15:     7085904      mithu      mithu         NA
16:     7096315     sonile     sonile         NA
17:     7137218    mostofa    mostofa         NA
18:  9908148515    merazol    merazol         NA
19: 99070511013     jaynal     jaynal         NA
```

```
print(idchk[hname.chk == 12 & exist == 123,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
            hhid hhh.name.1 hhh.name.2 hhh.name.3
  1:     7010105      monju      monju      montu
  2:     7010114  amiruddin  amiruddin amir uddin
  3:     7010115     anowar     anowar     anorar
  4:     7020205    sofikul    sofikul    sofiqul
  5:     7020206       alim       alim     rabeya
 ---
201: 98081710308   shohidul   shohidul     sokina
202:    81710511      belat      belat      amela
203:    81710516   moin ali   moin ali   moni ali
204:    81710520   tara mia   tara mia    taramia
205: 99081912105  hasen ali  hasen ali      hasen
```

```
print(idchk[hname.chk == 13 & exist == 123,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
Empty data.table (0 rows) of 4 cols: hhid,hhh.name.1,hhh.name.2,hhh.name.3
```

```
NULL
```

```
print(idchk[hname.chk == 13 & exist == 13,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
           hhid hhh.name.1 hhh.name.2 hhh.name.3
 1:     7021218    ebrahim         NA    ebrahim
 2:     7031815      kader         NA      kader
 3:     7054504 sohor vanu         NA sohor vanu
 4:     7064602 sona monsi         NA sona monsi
 5:     7064617       somu         NA       somu
---
 7:  9807133512     saiful         NA     saiful
 8:     7137305    mofidul    mofidul    mofidul
 9:     8159216     mokles         NA     mokles
10:    81710513     aminur         NA     aminur
11: 99081912406      rezia         NA      rezia
```

```
print(idchk[hname.chk == 23 & exist == 123,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
            hhid hhh.name.1   hhh.name.2   hhh.name.3
  1:     7010106      rongu        ronju        ronju
  2:     7010108   chan mia         chan         chan
  3:     7010117       ahad     ahad ali     ahad ali
  4:     7020214      asraf      mojiron      mojiron
  5:     7020220      yamal        jamal        jamal
 ---
387: 99081912109      bisha bisha mondol bisha mondol
388: 99081912115      sahar        sahan        sahan
389: 99081912118  fojor ali        fojor        fojor
390: 99081912413     satter       sattar       sattar
391: 99081912416    golijar       goljar       goljar
```

```
print(idchk[hname.chk == 23  & exist == 23,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
Empty data.table (0 rows) of 4 cols: hhid,hhh.name.1,hhh.name.2,hhh.name.3
```

```
NULL
```

```
print(idchk[hname.chk == 123  & exist == 123,
        grepout(”hh”, colnames(idchk)), with = F], nrow = 10)
```

```
             hhid hhh.name.1 hhh.name.2 hhh.name.3
   1:     7010101     haydar     haydar     haydar
   2:     7010102  soraf mia  soraf mia  soraf mia
   3:     7010107      rafiq      rafiq      rafiq
   4:     7010109       amir       amir       amir
   5:     7010110      golam      golam      golam
  ---
1149: 99081912411      kader      kader      kader
1150: 99081912412  tarifulla  tarifulla  tarifulla
1151: 99081912414     saiful     saiful     saiful
1152: 99081912415      golam      golam      golam
1153: 99081912420     sukkur     sukkur     sukkur
```

Do the non-typo changes in respondent names indicate that different households are given the same hhid? Below looks OK.

```
idchk[grepl(”kala chan”, hhh.name.1),
        grepout(”name”, colnames(idchk)), with = F]
```

```
   g.name rname.1 hhh.name.1 ename.1 sname.1 deo.name.1 rname.2 hhh.name.2
1:  hazra   hazra  kala chan   jasim   palas         NA      NA         NA
   ename.2 sname.2 deo.name.2 rname.3 hhh.name.3 ename.3 sname.3 deo.name.3
1:      NA      NA         NA  hajera hojrot ali   jasim  palash      bablu
   rname.chk hname.chk
1:         0         0
```

We must ask GUK why head names change across rounds. May be change in head? I will assume that, after correcting for duplication, hhids are correct and same households are visisted for the same hhid. We should have never dropped member names from the cover page.

III check treatment (lstatus and cover page information)

```
setwd(pathsave)
idfu ← readRDS(”idfu.rds”)
setkey(idfu, hhid)
```

III.1 arm

Found: Little discrepancy in arm consistency in the identification file. For surviving subjects, treatment assignment given in treatment in loan_status.dta matches with the original assignment given in treat of treatment assignment instruction file individual_treatment_assignment.prn. By rd 2, majority of the control started to receive treatments. It took about 250 days to complete the treatment on the treated, only then the treatment on the control started. Some of the individuals rejected to be treated stay in the sample.

III.2 contemporaneous treatment assignment

Contemporaneous (time-varying) treatment assignment, T/C, is stored in a separate file ./1/original/loan_status_october15-2015.prn.

```
setwd(pathsource.mar)
lstatus ← fread(grepout(”loan.status”, fn), integer64 = ”double”)
setnames(lstatus, ”groupid”, ”gid”)
```

Apply hhid correction of the previous subsection (Not shown here).

Create memstts from the membership status in lstatus file by following the same way as memstatus in cover page files (Not shown here).

Recode rand_arm in lstatus in the same way as in cover pages (Not shown here).

```
setwd(pathsave)
attrit ← fread(”attrition.prn”)
setnames(attrit, c(”i1”, ”i2”, ”i3”), paste(”inData”, 1:3, sep = ”.”))
attrit[, c(”inData.1”, ”inData.2”, ”inData.3”) := NULL]
setkey(attrit, hhid); setkey(lstatus, hhid)
lsa ← attrit[lstatus]
```

lsa: attrition (cover) + lstatus.

All members of gid 817112 are missing in cover page data while included in ./1/original/loan_status_october15-2015.prn.

```
lsa[is.na(exist), .(gid, hhid, memname, assignment, purchaseDate, memstts)]
```

```
       gid        hhid     memname assignment purchaseDate   memstts
 1:  71373  9997137316     monoara    treated   2013-10-06       old
 2:  71373  9997137317     afruja    control   2014-10-12       old
 3: 817112 99081711201     rohiton    control   2014-11-19 new group
 4: 817112 99081711202    monowara    treated   2013-11-05 new group
 5: 817112 99081711203      saheda    treated   2013-11-05 new group
 6: 817112 99081711204      sabina    treated   2013-11-05 new group
 7: 817112 99081711205     sahiron    control   2014-11-19 new group
 8: 817112 99081711206      sukzan    control   2014-04-20 new group
 9: 817112 99081711207       laily    control   2014-04-20 new group
10: 817112 99081711208      joshna    control   2014-04-20 new group
11: 817112 99081711209 samsunnahar    treated   2013-11-05 new group
12: 817112 99081711210     roshida    treated   2013-11-05 new group
13: 817112 99081711211     morjina    treated   2013-11-05 new group
14: 817112 99081711212    monjuara    treated   2013-11-05 new group
15: 817112 99081711213     sukiron    treated   2013-11-05 new group
16: 817112 99081711214     anowara    treated   2013-11-05 new group
17: 817112 99081711215    moshlima    treated   2013-11-05 new group
18: 817112 99081711216      jobeda    control   2015-03-07 new group
19: 817112 99081711217     kohinur    control   2015-03-07 new group
20: 817112 99081711218      maleka    control   2014-11-19 new group
21: 817112 99081711219      sokina    control   2015-05-25 new group
22: 817112 99081711220     sahinur    control   2014-11-19 new group
```

```
       gid         hhid      memname assignment purchaseDate    memstts
```

Need to inquire why these members are missing from data.

creditstatus gives the current treatment status. assignment gives the treatment status assigned at the beginning of rd 2. 1766 subjects were eventually treated by rd 3, of which 846 (47.9%) are the original control.

```
table0(lstatus[,.(creditstatus, assignment)])
```

```
            assignment
creditstatus control treated
         No     143      90
         Yes    846     920
```

90 did not accept the credit even though they were assigned to the original treated. These are individual rejecters. 143 original controls who stayed as the control are a mix of individual rejecters, new group, old, and replacement. The details will be explored later using idt.

Disaggregate by arms. Use lsa (cover pages & lstatus) to incorporate attrited subjects.

```
lsa[, creditstatus := asc(creditstatus)]
lsa[grepl(”^1?2?$”, exist), creditstatus := ”attrited”]
lsa[, creditstatus := factor(creditstatus, levels = c(”attrited”, ”No”, ”Yes”))]
#tb.treat.arm ← table(lstatus[,.(creditstatus, assignment, arm)])
tb.treat.arm ← table(lsa[,.(creditstatus, assignment, arm)])
arms ← levels(lsa[, arm])
tb.treat.arm ← lapply(1:dim(tb.treat.arm)[3],
        function(i) data.table(cbind(arm = arms[i],
        creditstatus = c(”attrited”, ”no”, ”yes”), tb.treat.arm[, , i])))
tb.treat.arm ← rbindlist(tb.treat.arm)
setnames(tb.treat.arm, colnames(tb.treat.arm),
        c(”arm”, ”receivedCredit”, ”originalControl”, ”originalTreated”))
tb.treat.arm ← rbindlist(
        list(cbind(assignment = ”control”,
        tb.treat.arm[, c(”arm”, ”receivedCredit”, ”originalControl”), with = F]),
        cbind(assignment = ”treated”,
        tb.treat.arm[, c(”arm”, ”receivedCredit”, ”originalTreated”), with = F])))
setnames(tb.treat.arm, ”originalControl”, ”value”)
tb.treat.arm[, value := asn(value)]
tb.treat.arm[, assignment := factor(assignment, levels = c(”control”, ”treated”))]
tb.treat.arm[, arm := factor(arm, levels = c(”traditional”, ”large”, ”large grace”, ”cow”)
tb.treat.arm[, receivedCredit := factor(receivedCredit, levels = c(”yes”, ”no”, ”attrited”
#tb.treat.arm[, receivedCredit := factor(receivedCredit,
#       levels = rev(levels(tb.treat.arm$receivedCredit)))]
```

(Programming memo: Setting key is neccesary for stacking order of bar plot. Don't know why it is the case, but changing the factor level ordering does not help.)

```
setkey(tb.treat.arm, assignment, arm, receivedCredit)
```

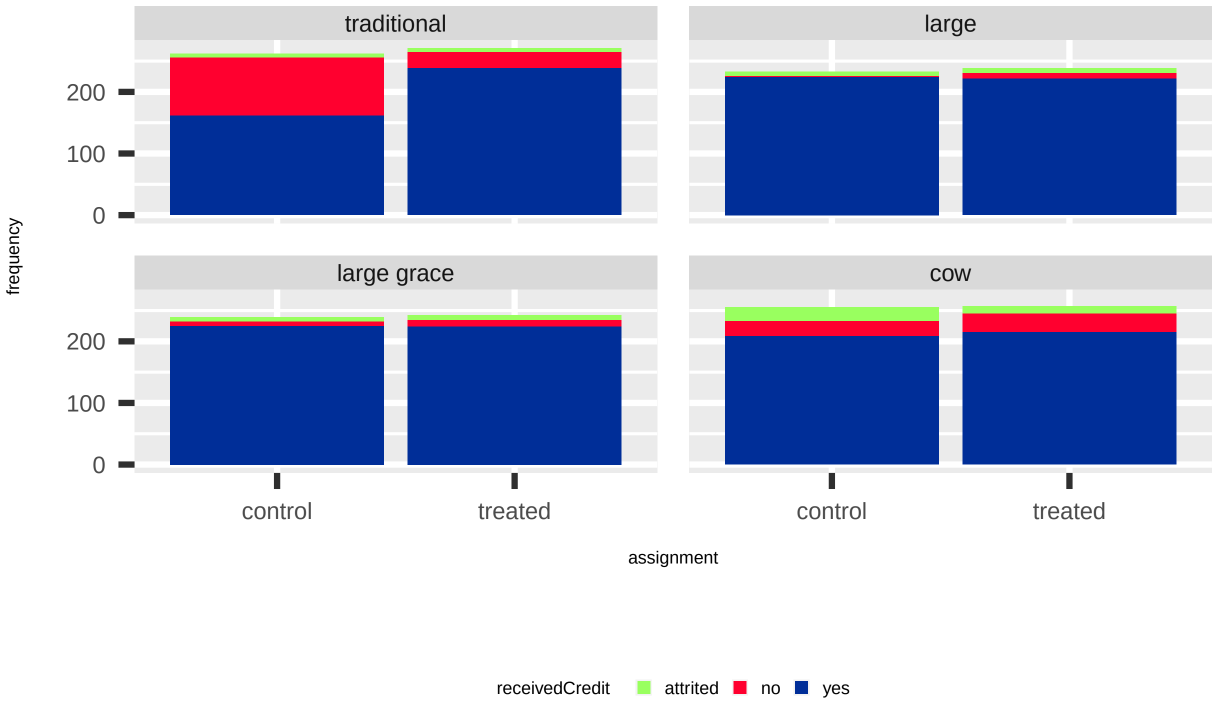
Figure 1 shows that large and large grace have smaller rejections. The controls in traditional arm has the highest rejections.

Check the dates of disbursement.

```
lstatus[, disburseDate := as.POSIXct(distributiondate, format = ”%d/%m/%Y”)]
lstatus[, purchaseDate := as.POSIXct(purchaseddate, format = ”%d/%m/%Y”)]
lstatus[, disburse.y := year(disburseDate)]
```

233 NAs in disbursement date are yet-to-be treated subects.

```
(tb ← table(lstatus[is.na(disburse.y), .(assignment, creditstatus)]))
```



Notes: The status is up to round 3 information. Attrition is defined as being not obsrved in round 3.

Figure 1 Treatment acceptance

```
          creditstatus
assignment  No Yes
   control 143   1
   treated  90  10
```

11 anomalies are typo’s. Dates are wrongly formated.

```
print(lstatus[is.na(disburse.y) & grepl(”Y”, creditstatus), distributiondate], quote = F)
```

```
 [1] 12512/2013 30/102013  30/102013  30/102013  30/102013  30/102013
 [7] 30/102013  30/102013  30/102013  30/102013  30/102013
```

Correct date typo’s.

```
lstatus[is.na(disburse.y) & grepl(”Y”, creditstatus),
        distributiondate := gsub(”12512”, ”12/12”, distributiondate)]
lstatus[is.na(disburse.y) & grepl(”Y”, creditstatus),
        distributiondate := gsub(”1020”, ”10/20”, distributiondate)]
```

After correcting for typo’s, 233 cases of no dates are subjects who did not receive a credit. Disbursement months and weeks are widely distributed.

```
table0(lstatus[, .(disburse.y, disburse.m)])
```

```
          disburse.m
disburse.y   1   2   3   4   5   6   7   8   9  10  11  12 <NA>
      2013   0   0   3 211 198 137  11   1  86 323  28   6    0
      2014  13  15  46  76   7   0   8   1   0 185  60  81    0
      2015   0  42  76   0  48   0   0 104   0   0   0   0    0
      <NA>   0   0   0   0   0   0   0   0   0   0   0   0  233
```

```
table0(lstatus[, .(disburse.y, disburse.w)])
```

```
          disburse.w
disburse.y    1   2   3   4   5   6   7   8   9  10  11  12  13  14  16  17  18
      2013    0   0   0   0   0   0   0   0   0   0   0   3   0   2  84 105  40
      2014    4   1   2   2   6  12   0   0   1   8  15  20   3   8  19  47   7
      2015    0   0   0   0   8   4  24   6  23   0  53   0   0   0   0   0  20
      <NA>    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          disburse.w
disburse.y   19  20  21  22  23  24  25  26  28  29  30  31  32  33  34  36  37
      2013    7  22  75  74  16 105  14   2   0  11   0   0   0   1   0  17  10
      2014    0   0   2   0   0   0   0   0   1   0   7   0   0   0   1   0   0
      2015   13  15   0   0   0   0   0   0   0   0   0  36  53  15   0   0   0
      <NA>    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          disburse.w
disburse.y   38  39  40  41  42  43  44  45  46  47  48  50  51  52 <NA>
      2013   26  32  87  52  42  57  86  12  13   2   1   1   2   3    0
      2014    0   0  90   0  95   0   0  60   0   0   0  81   0   0    0
      2015    0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
      <NA>    0   0   0   0   0   0   0   0   0   0   0   0   0   0  233
```

```
#table(lstatus[, .(disburse.y, disburse.m, arm)], useNA = ”ifany”)
```

Compute elapsed treatment days up to Oct 1, 2015 by original treatment assignment.

```
lstatus[, elapsed := as.POSIXct(”10/01/2015”, format = ”%m/%d/%Y”) − disburseDate]
```

Summary of elapsed days by assignment.

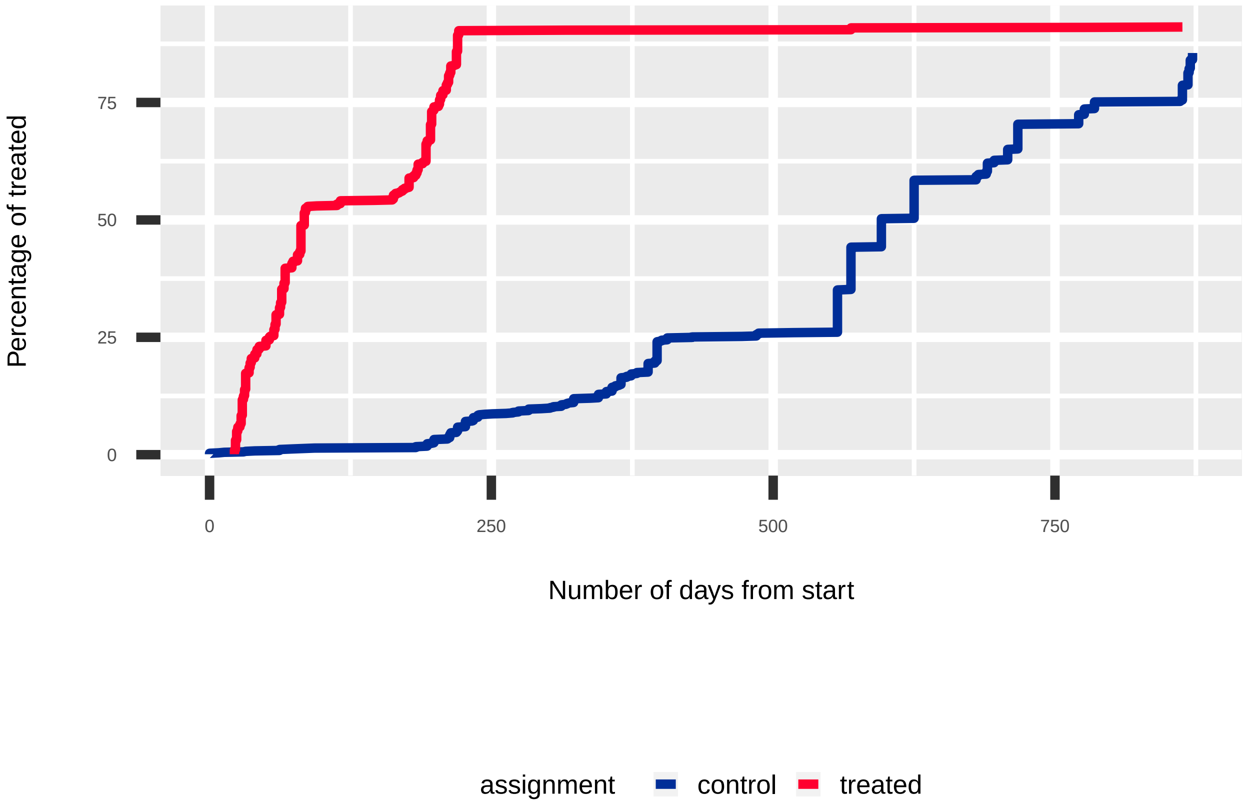
```
el.by.D ← by(lstatus[, elapsed], lstatus[, assignment], destat, prob = ”odd”)
el.by.D2 ← rbindlist(lapply(1:length(el.by.D),
        function(i) data.table(cbind(levels(lstatus[, assignment])[i], t(asn(el.by.D[i]))))
setnames(el.by.D2, colnames(el.by.D2), c(”assignment”, colnames(el.by.D[[1]])))
el.by.D2
```

```
   assignment min 10\\% 25\\% median 75\\% 90\\% max  mean   std 0s NAs    n
1:    control  49     58   213    352   524 673.5 921 357.7 199.2  0 143  989
2:    treated  58    707   725    840   877   892 899 804.4  87.8  0  90 1010
```

Members were eventually given a treatment in our stepped wedge design. We can plot the rate of the treated against elapsed days in Figure 2, 3.

```
lstatus[, daysFromStart :=
        asn(disburseDate − min(disburseDate, na.rm = T))/(3600*24)]
# daysFromStart = NA is never-treated subject
#  cumulative treatment rate by treatment assignment
setkey(lstatus, assignment, daysFromStart)
lstatus[, cumTreated := as.double(cumsum(grepl(”Y”, creditstatus))), by = assignment]
lstatus[, en := .N, by = assignment]
lstatus[, cumTreated := cumTreated/en]
lstatus[, en := NULL]
#  cumulative treatment rate by treatment assignment and arm
lstatus[, cumTreatedA := as.double(cumsum(grepl(”Y”, creditstatus))),
        by = c(”assignment”, ”arm”)]
lstatus[, en := .N, by = c(”assignment”, ”arm”)]
lstatus[, cumTreatedA := cumTreatedA/en]
lstatus[, en := NULL]
setkey(lstatus, assignment, daysFromStart)
```

Disbursement time lag within the village by the group. In Figure 4, there are 3 villages (81693, 703114, 703115) with no control receiving credits. There are a few villages where the controls precede some of the treated. Probably miscoding in data capture.



Notes: The cumulative total of receiving treatment among the respective original treatment assignment groups defined in ’assignment’.

Figure 2 Treatment progression

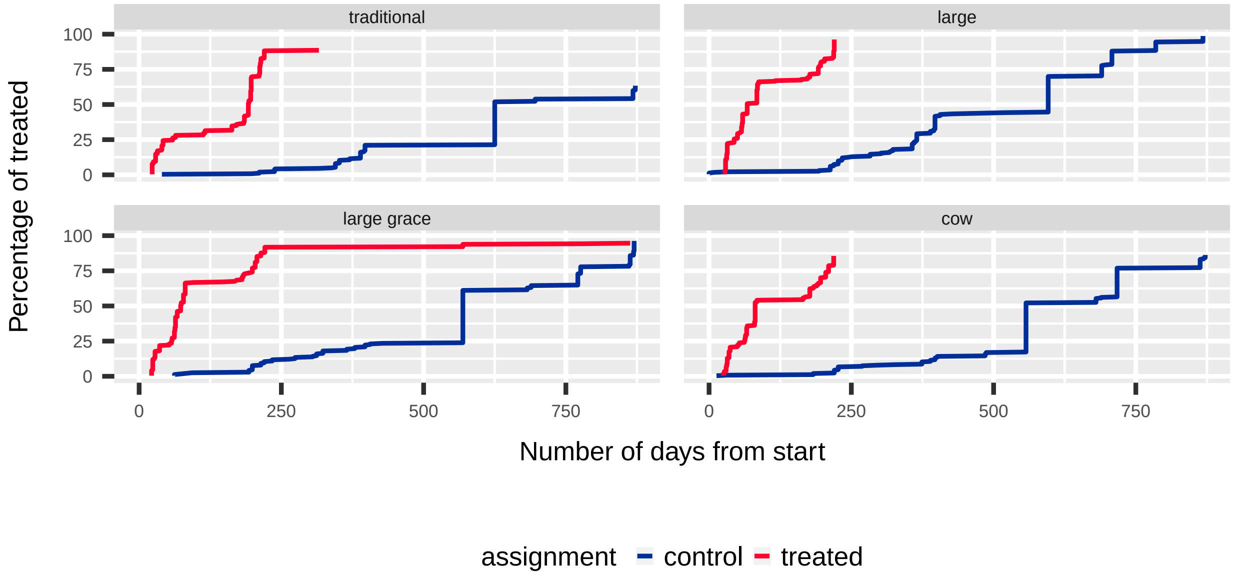


Figure 3 Treatment progression by arm

```
setkey(lstatus, gid, daysFromStart)
lstatus[, daysFromStartV := daysFromStart − min(daysFromStart, na.rm = T), by = gid]
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
Warning: Removed 233 rows containing non-finite values (stat_bin).
```

Check 3 villages (81693, 703114, 703115). Controls have NAs in disburseDate but membership status is not individual rejecters. Forgot to capture disbursement dates?

```
table(lstatus[(gid == 81693 | gid == 703114 | gid == 703115) &
        grepl(”co”, assignment), .(memstts, disburseDate)], useNA = ”ifany”)
```

```
           disburseDate
memstts     <NA>
  new group   20
  old         10
```

IV original vs. contemporaneous treatment assignment (lstatus and randomisation information)

IV.1 original vs. contemporaneous treatment assignment

Check assignment against our instructions on treatment assignment. Read treatment assigment from individual treatment assignment files in randomization folder. Arms by individual are stored in ./Randomization/Randomization/individual_treatment_assignment.prn. Arms by char are stored in ./Randomization/Randomization/char_arm_assignment.prn. Merge individual treatmet status to original treatment assignment data to check consistency between the two.

```
setwd(pathreceived.oct)
ta ← list.files(pattern = ”.prn$”, recursive = T, full.names = T)
ta ← ta[!grepl(”fwd”, ta)]
Ta ← lapply(ta, fread, integer64 = ”double”)
```

treat is treatment assignment that we instructed to the field team. assignment is the treatment assignment recorded in the field.

```
table0(Ta[[2]][, treat])
```

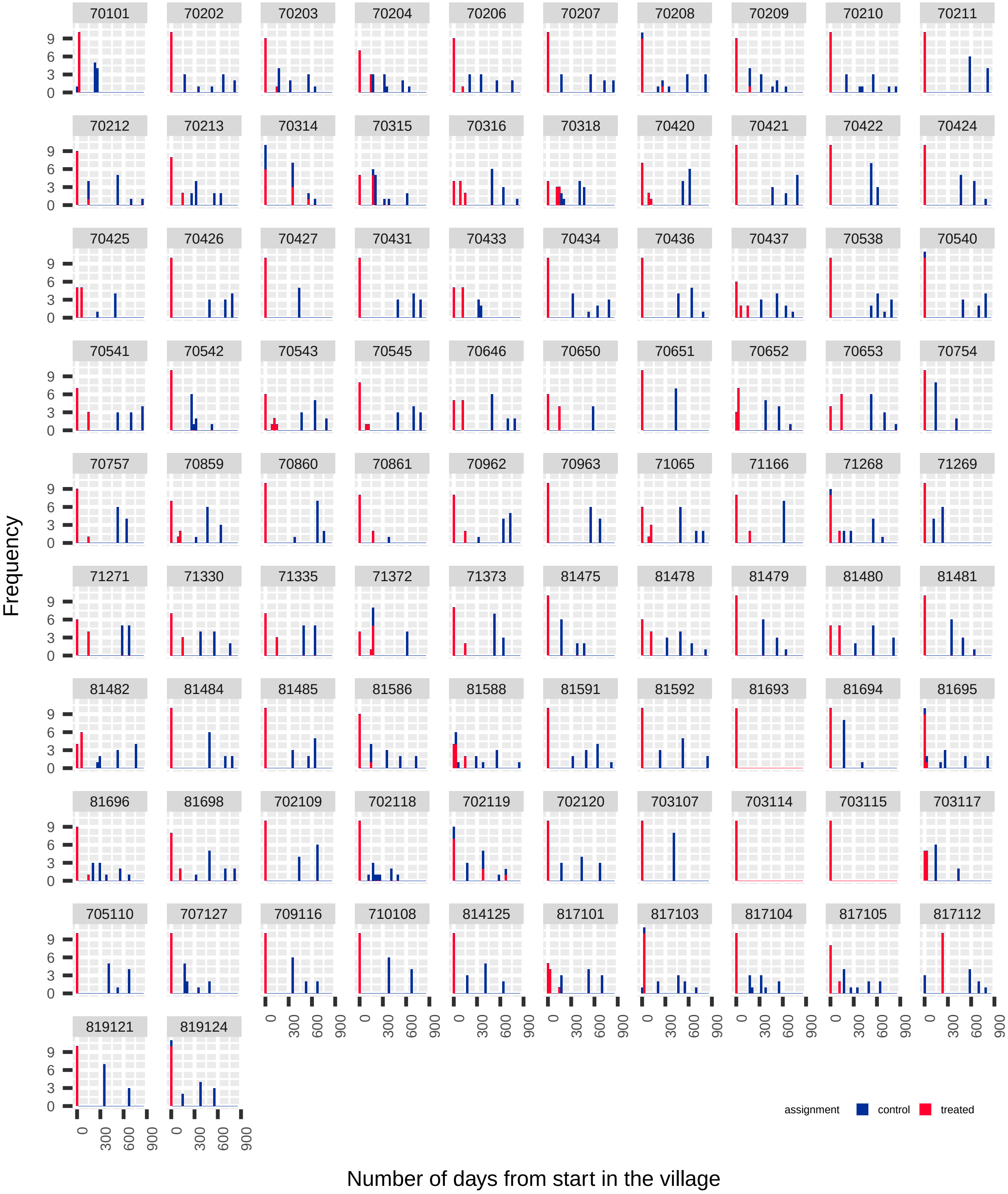
```

control treated
    800     800
```

Merge two files.

```
setnames(Ta[[2]], ”hh_id”, ”hhid”)
Ta[[2]][, treat := factor(treat)]
# as.double is necessary for merge in data.table
lstatus[, hhid := as.double(hhid)]
Ta[[2]][, hhid := as.double(hhid)]
setkey(lstatus, hhid); setkey(Ta[[2]], hhid)
tl ← Ta[[2]][lstatus]
```

tl is derived from lstatus and ./Randomization/Randomization/individual_treatment_assignment.prn. tl: lstatus + randomisation.



Notes: Treatment assignment according to the field record. Number of days since the treatment started in a village. Villages 81693, 703114, 703115 have missing dates.

Figure 4 Within village disbursement time gap

Below tabulation shows that our original instruction of treatment assignment is followed strictly for the surviving subjects. There are 621 subjects who were not present at the time of original treatment assignment randomisation. These subjects do not have UP/MP classification. There are NAs in treat because its source (./Randomization/Randomization/individual_treatment_assignment.prn) does not contain additional households from new chars (requested to Abu-san to add hhid to the randomization file, 2017 Mar 5).

```
table(tl[, .(assignment, treat)], useNA = ”ifany”)
```

```
          treat
assignment control treated <NA>
   control     689       0  300
   treated       0     689  321
```

treat is an indicator variable for everyone in the group. We stratified the subjects into ultra poor (UP) and moderate poor (MP), and assigned 7 out of 14 UPs and 3 out of 6 MPs to treatment. So a half of the subjects are classified as control/treated in treat.

```
table(tl[, .(assignment, treat, poverty.status)], useNA = ”ifany”)
```

```
, , poverty.status = mp

          treat
assignment control treated <NA>
   control     205       0    0
   treated       0     205    0

, , poverty.status = up

          treat
assignment control treated <NA>
   control     484       0    0
   treated       0     484    0

, , poverty.status = NA

          treat
assignment control treated <NA>
   control       0       0  300
   treated       0       0  321
```

The original treated received treatments in 2013. 90 opted out not to receive loans despite originally being selected as the treated.

```
table0(tl[, .(disburse.y, treat)])
```

```
          treat
disburse.y control treated <NA>
      2013      80     598  326
      2014     311       1  180
      2015     189       0   81
      <NA>     109      90   34
```

```
table0(tl[, .(creditstatus, treat)])
```

```
            treat
creditstatus control treated <NA>
          No     109      90   34
         Yes     580     599  587
```

IV.2 cover page and original randomisation arm assignment

```
idmem ← idfu[rd == last.rd, .(gid, hhid, exist, memstatus, arm, rejOfGroup, rejInGroup)]
```

```
setkey(tl, gid, hhid); setkey(idmem, gid, hhid)
idt ← tl[idmem]
```

idt: tl (lstatus + randomisation) + idfu (cover page). idt's hhid uses cover page as a base, and there are 222 subjects who show up in cover page but not in lstatus. Among them, 2 are old members.

```
idt[is.na(assignment) & grepl(”old”, memstatus),
        .(memstatus, arm, gid, hhid, villagename, exist, assignment)]
```

```
   memstatus arm   gid    hhid villagename exist assignment
1:       old  NA 71373 7137316          NA   123         NA
2:       old  NA 71373 7137317          NA   123         NA
```

Define memstatus, receivedCredit.

```
idt[(rejOfGroup), memstatus := ”group rejection”]
idt[, memstatus := factor(memstatus)]
idt[grepl(”Ul”, povertystatus), poverty.status := ”up”]
idt[grepl(”Mo”, povertystatus), poverty.status := ”mp”]
idt[!is.na(i.arm) & is.na(arm), arm := i.arm]
idt[, receivedCredit := NA]
#idt[grepl("^1?2?$", exist), receivedCredit := F]
idt[grepl(”Y”, creditstatus), receivedCredit := T]
idt[grepl(”N”, creditstatus), receivedCredit := F]
# not receiving credit if before intervention, lost to flood, group rejection
idt[is.na(receivedCredit) & grepl(”lost”, memstatus), receivedCredit := F]
idt[is.na(receivedCredit) & grepl(”gro.∗rej”, memstatus), receivedCredit := F]
#idt[is.na(receivedCredit), .(receivedCredit, memstatus, arm, gid, hhid, villagename, exis
```

Rejections come at a single stage. Despite we randomise at two levels, group and individual, we announced the results after both stages were complete. Groups and individuals rejected after learning the arm and inital treatment assignment status. Create an indicator of accept/reject rather than embedding it in arm or memstatus.

```
idt[, accept := ”yes”]
idt[grepl(”ind.∗rej”, memstatus), accept := ”individual rejection”]
idt[grepl(”gr.∗rej”, memstatus), accept := ”group rejection”]
```

assignment information is not shown for some subjects.

```
table0(idt[, .(assignment, memstatus)])
```

```
          memstatus
assignment old replacement new group individual rejection group rejection
   control 620          69       210                    69              10
   treated 599          90       210                    90              10
   <NA>      2           0         0                     0             140
          memstatus
assignment lost to flood
   control             0
   treated             0
   <NA>               80
```

Some of flooded individuals are under arm "before intervention". Need to change to "lost to flood".

```
table0(idt[grepl(”ye”, accept) & grepl(”flo”, memstatus), .(arm, memstatus)])
```

```
                         memstatus
arm                       lost to flood
  traditional                         0
  large                               0
  large grace                         0
  cow                                 0
  before intervention                24
  lost to flood                      56
```

```
idt[grepl(”ye”, accept) & grepl(”flo”, memstatus) & grepl(”bef”, arm),
        arm := ”lost to flood”]
table0(idt[grepl(”ye”, accept) & grepl(”flo”, memstatus), .(arm, memstatus)])
```

```
                         memstatus
arm                       lost to flood
  traditional                         0
  large                               0
  large grace                         0
  cow                                 0
  before intervention                 0
  lost to flood                      80
```

2 cases who accepted but quit. (Same 2 cases who has NA in receivedCredit)

```
idt[grepl(”ye”, accept) & is.na(assignment) & grepl(”cow”, arm),
        .(memstatus, arm, gid, hhid, villagename, exist, receivedCredit, accept)]
```

```
   memstatus arm   gid    hhid villagename exist receivedCredit accept
1:       old cow 71373 7137316          NA   123             NA    yes
2:       old cow 71373 7137317          NA   123             NA    yes
```

Change them to accept = individual rejection.

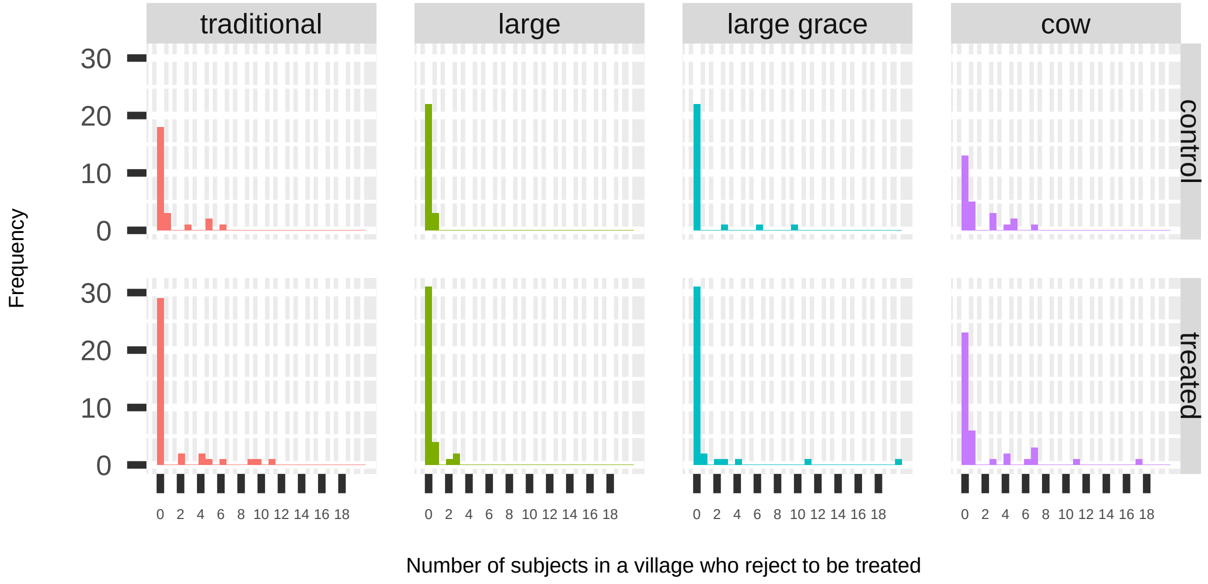
```
idt[grepl(”ye”, accept) & is.na(assignment) & grepl(”cow”, arm),
        accept := ”individual rejection”]
```

We examine the number of rejecters by arm and assignment. (Using accept or memstatus does not make a difference because the former is defined with the latter variable.) Compare with individual or group rejecters after excluding flood:

```
table0(idt[grepl(”rej”, memstatus) & !grepl(”flood”, memstatus) &
        grepl(”tr”, assignment), .(assignment, accept)])
```

```
          accept
assignment group rejection individual rejection
   control              10                   69
   treated              10                   90
```

```
nc1 ← data.table(table0(idt[grepl(”rej”, memstatus) & grepl(”co”, assignment), .(gid, arm
nc2 ← data.table(table0(idt[grepl(”rej”, memstatus) & grepl(”tr”, assignment), .(gid, arm
# if lost to flood, no assignmemt
nc3 ← data.table(table0(idt[grepl(”fl”, memstatus), .(gid, arm)]))
acpt ← rbind(
        cbind(assignment = ”control”, flood = F, nc1),
        cbind(assignment = ”treated”, flood = F, nc2))
acpt[, arm := factor(arm, levels = c(”traditional”, ”large”, ”large grace”, ”cow”))]
acpt[, assignment := factor(assignment, levels = c(”control”, ”treated”))]
```



Notes: The status is up to round 3 information. Number of subjects who chose to stay as the control.
Top panels are original control, bottom panels are original treated. Observation per village.

Figure 5 Treatment rejections in a group

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

FIGURE 5 shows that traditional and cow have more treatment rejecters. The controls in cow arm has the highest rejections.

IV.3 save treatment related information file

```
newcolnames ← c(”district”, ”upazila”, ”union”, ”village”,
          ”gid”, ”hhid”, ”mname”, ”husFatherName”,
          ”povstatus”, ”memstatus”, ”receivedCredit”, ”assignment”, ”arm”,
          ”accept”, ”disburseDate”, ”elapsed”, ”daysFromStart”,
          ”loanAmount”, ”iga1”, ”iga2”, ”iga3”, ”purchaseDate”)
setnames(idt, c(”districtname”, ”upazila”, ”unionname”, ”villagename”,
          ”gid”, ”hhid”, ”memname”, ”husbandfathersname”,
          ”poverty.status”, ”memstatus”, ”receivedCredit”, ”assignment”, ”arm”,
          ”accept”, ”disburseDate”, ”elapsed”, ”daysFromStart”,
          ”loananount”, ”IGA1”, ”IGA2”, ”IGA3”, ”purchaseDate”),
          newcolnames)
```

Note that the observations starting with 980/990 are individual replacement or group replacement. E.g., 9807020405 is replacement for 7020405.

```
table0(idt[grepl(”^9.0”, hhid), memstatus])
```

```
    replacement         new group  group rejection
            159               420               20
```

```
table0(idt[hhid %in% gsub(”^9.0”, ””, idt[grepl(”^9.0”, hhid), hhid]), memstatus])
```

```
individual rejection      group rejection
                 159                    20
```

```
summary(idt[, newcolnames, with = F])
```

```
   district           upazila              union              village
 Length:2199        Length:2199        Length:2199        Length:2199
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character




      gid              hhid             mname              husFatherName
 Min.   : 70101   Min.   :7.01e+06   Length:2199        Length:2199
 1st Qu.: 70426   1st Qu.:7.04e+06   Class :character   Class :character
 Median : 70860   Median :7.13e+06   Mode  :character   Mode  :character
 Mean   :198132   Mean   :1.51e+10
 3rd Qu.: 81591   3rd Qu.:9.81e+09
 Max.   :819124   Max.   :9.91e+10

 povstatus                  memstatus    receivedCredit    assignment
 mp  : 600   old                 :1221    Mode :logical   control:978
 up  :1377   replacement         : 159    FALSE:453       treated:999
 NA's: 222   new group           : 420    TRUE :1744      NA's   :222
             individual rejection: 159    NA's :2
             group rejection     : 160
             lost to flood       :  80

                  arm                       accept
 traditional        :611   group rejection     : 160
 large              :492   individual rejection: 161
 large grace        :502   yes                 :1878
 cow                :512
 before intervention:  2
 lost to flood      : 80

  disburseDate                    elapsed         daysFromStart   loanAmount
 Min.   :2013-04-01 00:00:00   Length:2199       Min.   :  8   Min.   :    0
 1st Qu.:2013-06-10 00:00:00   Class :difftime   1st Qu.: 78   1st Qu.: 7840
 Median :2013-10-24 00:00:00   Mode  :numeric    Median :214   Median :16800
 Mean   :2014-02-17 10:20:05                     Mean   :330   Mean   :12708
 3rd Qu.:2014-10-14 00:00:00                     3rd Qu.:569   3rd Qu.:16800
 Max.   :2015-08-13 00:00:00                     Max.   :872   Max.   :16800
 NA's   :455                                     NA's   :455   NA's   :222
            iga1                    iga2                    iga3
 cow rearing  :1308                   :1948                   :1948
              : 245   buying machine  :   1   cow rearing     :   4
 small trade  : 190   cow rearing     :   3   land mortgage   :  11
 goat rearing : 103   land mortgage   :  15   nut,corn farming:  13
 land mortgage:  84   nut,corn farming:  10   oxe rearing     :   1
 (Other)      :  47   NA's            : 222   NA's            : 222
 NA's         : 222
  purchaseDate
 Min.   :2013-02-01 00:00:00
 1st Qu.:2013-06-18 00:00:00
 Median :2013-10-26 00:00:00
 Mean   :2014-02-19 11:28:35
 3rd Qu.:2014-10-24 00:00:00
```

```
Max.    :2015-08-15 00:00:00
NA's    :480
```

```
setwd(pathsave)
tr ← idt[, newcolnames, with = F]
write.tablev(tr, ”treatment_assignment.prn”)
saveRDS(tr, ”treatment_assignment.rds”)
```

tr: idt (tl (lstatus + randomisation) + idfu (cover page)) with most consistent arm, assignment.

IV.4 interview dates

Found: Some dates are erroneously entered. 21 cass of missing interview dates. Interview dates. (Correct some typos before date conversion.)

```
idfu[rd == 1, intDate := as.POSIXct(asc(st.day.inter), format = ”%d−%m−%Y”)]
# correct typo 1
idfu[rd == 1 & is.na(intDate) & grepl(”092”, st.day.inter),
        st.day.inter := gsub(”092”, ”09/2”, st.day.inter)]
# correct typo 2
idfu[rd == 1 & is.na(intDate) & grepl(”;13”, st.day.inter),
        st.day.inter := gsub(”;”, ””, st.day.inter)]
idfu[rd == 1, intDate := as.POSIXct(asc(st.day.inter), format = ”%d−%m−%Y”)]
idfu[rd == 1 & is.na(intDate) & grepl(”−”, st.day.inter),
        intDate := as.POSIXct(asc(st.day.inter), format = ”%d−%m−%y”)]
idfu[rd == 1 & is.na(intDate) & grepl(”\\/”, st.day.inter),
        intDate := as.POSIXct(asc(st.day.inter), format = ”%d/%m/%Y”)]
idfu[rd ≥ 2, intDate := as.POSIXct(asc(d.1st), format = ”%d%m%y”)]
idfu[rd ≥ 2 & is.na(intDate) & !is.na(st.day) & !is.na(st.month) & !is.na(st.year),
        intDate := as.POSIXct(paste0(st.day, ”−”, st.month, ”−”, st.year), format = ”%d−%m
```

Incorporate disburseDate to show the timing of intervention in terms of rd.

```
iddates ← idfu[, .(gid, hhid, rd, intDate, memstatus)]
disdates ← lstatus[, .(gid, hhid, disburseDate, purchaseDate)]
setkey(iddates, gid, hhid); setkey(disdates, gid, hhid)
iddates ← disdates[iddates]
iddates[, disbursed := NA]
iddates[disburseDate > intDate, disbursed := F]
iddates[disburseDate ≤ intDate, disbursed := T]
```

In rd 2 onwards, subjects who quit before C/T assignment has disburse = NA. Turn this to disburse = F.

```
iddates[, purchased := NA]
iddates[purchaseDate > intDate, purchased := F]
iddates[purchaseDate ≤ intDate, purchased := T]
setkey(iddates, hhid, rd)
iddates[, disbursed1 := shift(disbursed, 1L, type=”lag”), by = hhid]
iddates[, purchased1 := shift(purchased, 1L, type=”lag”), by = hhid]
```

If disbursed/purchased in previous round, later round is T.

```
iddates[is.na(disbursed) & !is.na(disbursed1) & disbursed1,
        disbursed := disbursed1]
iddates[is.na(purchased) & !is.na(purchased1) & purchased1,
        purchased := purchased1]
iddates[, c(”disbursed1”, ”purchased1”) := NULL]
```

For rd 1, group rejection, individual rejection, and lost to flood, there is no disbursement.

```
iddates[rd == 1 | grepl(”reje|lost”, memstatus),
        c(”disbursed”, ”purchased”) := list(F, F)]
```

Errors in year: not 0011, 0012, 2004, 2005, 2011. Correct them to 2012, 2012, 2014, 2015, 2012, respectively. **Is revision correct**?

```
print(unique(iddates[year(intDate) ≤ 2010, year(intDate)]), quote = F)
```

```
[1]   12 2004 2005   11
```

```
iddates[year(intDate) == 11, .(gid, hhid, rd, intDate, disburseDate)]
```

```
      gid     hhid rd    intDate disburseDate
1: 817103 81710303  1 0011-10-09   2013-04-23
2: 817103 81710304  1 0011-10-09   2013-04-23
3: 817103 81710305  1 0011-10-14   2015-03-11
4: 817103 81710307  1 0011-10-09   2013-04-23
5: 817103 81710308  1 0011-10-09         <NA>
```

There are subjects whose interview dates are missing in rd 1.

```
table0(iddates[is.na(year(intDate)), .(gid, rd)])
```

```
       rd
gid      1  2
  70314 19  0
  70437  0  1
  70963  1  0
```

Mostly the group rejected households.

```
table0(iddates[is.na(year(intDate)), .(rd, memstatus)])
```

```
   memstatus
rd  old group rejection
  1   1              19
  2   1               0
```

```
#<<eval = F>≥
invisible(iddates[year(intDate) == 2005,
        intDate := as.POSIXct(paste(”2015”, month(intDate), mday(intDate), sep = ”−”),
        format = ”%Y−%m−%d”)])
invisible(iddates[year(intDate) == 2004,
        intDate := as.POSIXct(paste(”2014”, month(intDate), mday(intDate), sep = ”−”),
        format = ”%Y−%m−%d”)])
invisible(iddates[year(intDate) == 11 | year(intDate) == 2011 | year(intDate) == 12,
        intDate := as.POSIXct(paste(”2012”, month(intDate), mday(intDate), sep = ”−”),
        format = ”%Y−%m−%d”)])
```

Check correction.

```
table0(iddates[, year(intDate)])
```

```

2012 2013 2014 2015 2016 <NA>
1578  618 2080 1550  542   21
```

For disbursement, we know that it should be before the first day of rd 3. If receivedCredit == T and rd 3, even intDate is NA, we know that disbursed/purchased == T. To do so, we need treatment information in tl.

```
setwd(pathsave)
tr ← fread(”treatment_assignment.prn”, integer64 = ”double”)
tr ← tr[, .(hhid, receivedCredit)]
setkey(tr, hhid)
```

```
setkey(iddates, hhid)
ivdates ← iddates[tr]
ivdates[rd == 3 & receivedCredit & is.na(disbursed), disbursed := T]
ivdates[rd == 3 & receivedCredit & is.na(purchased), purchased := T]
ivdates[rd == 3 & !receivedCredit & is.na(disbursed), disbursed := F]
ivdates[rd == 3 & !receivedCredit & is.na(purchased), purchased := F]
ivdates[(is.na(disbursed) & !is.na(disburseDate) & !is.na(intDate)) |
        (is.na(disbursed) & !is.na(disburseDate) & rd == 3) , ]
```

```
Empty data.table (0 rows) of 10 cols: gid,hhid,disburseDate,purchaseDate,rd,intDate...
```

```
table0(ivdates[is.na(disburseDate), .(rd, receivedCredit)])
```

```
   receivedCredit
rd   FALSE <NA>
  1    453    2
  2    371    2
  3    371    2
```

Remaining NAs in disburse are 2 obs.

```
ivdates[is.na(disburseDate) & is.na(receivedCredit), ]
```

```
     gid    hhid disburseDate purchaseDate rd    intDate memstatus disbursed
1: 71373 7137316         <NA>         <NA>  1 2012-10-05       old     FALSE
2: 71373 7137316         <NA>         <NA>  2 2014-01-21       old        NA
3: 71373 7137316         <NA>         <NA>  3 2015-11-11       old        NA
4: 71373 7137317         <NA>         <NA>  1 2012-10-05       old     FALSE
5: 71373 7137317         <NA>         <NA>  2 2014-01-21       old        NA
6: 71373 7137317         <NA>         <NA>  3 2015-11-21       old        NA
   purchased receivedCredit
1:     FALSE             NA
2:        NA             NA
3:        NA             NA
4:     FALSE             NA
5:        NA             NA
6:        NA             NA
```

```
table(ivdates[is.na(disbursed), .(memstatus, receivedCredit, rd)], useNA = ”ifany”)
```

```
, , rd = 2

                      receivedCredit
memstatus               FALSE TRUE <NA>
  old                      41    1    2
  replacement               6    0    0
  new group                28    0    0
  individual rejection      0    0    0
  group rejection           0    0    0
  lost to flood             0    0    0

, , rd = 3

                      receivedCredit
```

```
memstatus                FALSE TRUE <NA>
  old                        0    0    2
  replacement                0    0    0
  new group                  0    0    0
  individual rejection       0    0    0
  group rejection            0    0    0
  lost to flood              0    0    0
```

ivdates only shows the observed rds. Incorporate attrition information in iu to explicitly show the unobserbed rounds.

```
setwd(pathsave)
attrit ← fread("attrition.prn")
setnames(attrit, c("i1", "i2", "i3"), paste("inData", 1:3, sep = "."))
attrit ← reshape(attrit, direction = "long", idvar = "hhid",
        varying = paste("inData", 1:3, sep = "."))
setnames(attrit, "time", "rd")
setkey(ivdates, hhid, rd)
setkey(attrit, hhid, rd)
itvdates ← ivdates[attrit]
```

```
setwd(pathsave)
write.tablev(itvdates, "interview_dates_long.prn")
saveRDS(itvdates, "interview_dates_long.rds")
itvdatesw ← reshape(itvdates, direction = "wide",
        idvar = c("hhid", "exist"),
        timevar = "rd", v.names = grepout("ntD|ed$|inD", colnames(iddates)))
write.tablev(itvdatesw, "interview_dates_wide.prn")
saveRDS(itvdatesw, "interview_dates_wide.rds")
```

V food consumption and vulnerability

Found: Missing observations match with attrition. No problem found.
Let us match against Section 3B (Food consumption and vulnerability).

```
grepout("sec.*\\_3b", fn)
```

```
[1] "./2/section_3b.prn" "./3/section_3b.prn"
```

```
setwd(pathsource.mar)
sec3b ← lapply(grepout("sec.*\\_3b", fn), fread, integer64 = "double")
idunion3b ← unique(asn(lapply(sec3b, function(x) x[, id])))
idunion3b ← idunion3b[order(idunion3b)]
c(length(idunion), length(idunion3b))
```

```
[1] 2221 2125
```

```
table(idunion %in% idunion3b)
```

```

FALSE  TRUE
  120  2101
```

```
table(idunion3b %in% idunion)
```

```
FALSE  TRUE
   24  2101
```

```
for (i in 1:2) assign(paste0("i", i+1), idunion %in% sec3b[[i]][, id])
iu3b ← data.table(idunion, i1 = idunion %in% idunion3b, i2, i3)
iu3b[, exist := ""]
iu3b[(i1), exist := "0"]
iu3b[(i2), exist := paste0(exist, 2)]
iu3b[(i3), exist := paste0(exist, 3)]
iu3b[exist == "", exist := NA]
iu3b[, c("i1", "i2", "i3") := NULL]
iu3b[, exist := factor(exist, levels = c("0", "023", NA))]
setnames(iu3b, "idunion", "hhid")
```

195 NA's (almost) match with subjects who are observed up to two rounds (172 in total), NA matches with remainder of attrition last seen in rd 2.

```
table0(iu3b[, exist])
```

```
 023 <NA>
2026  195
```

VI rd1: merge original and additional samples

List files.

```
setwd(pathsource.mar)
foldername ← list.dirs(path = "./1", recursive = T, full.names = T)
foldername ← foldername[!grepl("^\\.\\/1$|combined", foldername)]
fn ← list.files(path = foldername, pattern = ".prn$",
        recursive = T, full.names = T)
fn ← unique(fn)
fn1 ← grepout("ad.*Sec", fn)
fn1 ← gsub("^.*\\/", "", fn1)
fn1 ← gsub(".prn", "", fn1)
print(fn1, quote = F)
```

```
 [1] Section_01  Section_10  Section_11  Section_12  Section_13  Section_14a
 [7] Section_14b Section_15a Section_15b Section_16  Section_17  Section_18
[13] Section_19  Section_20  Section_21  Section_22  Section_23a Section_23b
[19] Section_24  Section_2a  Section_2b  Section_2c  Section_3a  Section_3b
[25] Section_4a  Section_4b  Section_5a  Section_5b  Section_6a  Section_6b
[31] Section_7a  Section_8a  Section_8b  Section_9
```

```
fn1 ← substr(fn1, 2, nchar(fn1))
fn1 ← gsub("n\\_", "n", fn1); fn0 ← gsub("n\\_", "n", fn)
fn.0 ← gsub("ection", "s", fn1)
fn.0 ← gsub("s(\\d)(\\D)?$", "s0\\1\\2", fn.0)
```

Read files.

```
setwd(pathsource.mar)
R ← lapply(fn, fread, integer64 = "double", header = T)
```

Start from 2nd, as there is no ection01 file in original files.

```
duph ← NULL
for (i in 2:length(fn1)) {
        x1 ← R[grep(fn1[i], fn0)]
        x1 ← lapply(x1, function(x)
        if (any(grepl(”hh_id”, colnames(x)))) setnames(x, ”hh_id”, ”hhid”) else x)
        x1 ← rbindlist(x1, fill = T)
        x1 ← a2b(x1, ”.”, NA)
        x1 ← x1[!duplicated(x1), ]
        table(tb ← table0(x1[, .(hhid, mid)]))
        duph ← c(duph, length(duphhid ← rownames(tb[apply(tb > 1, 1, any), ])))
        setkey(x1, hhid, mid)
        write.tablev(x1, paste0(pathsource.mar, ”1/combined/”, fn.0[i], ”.prn”))
}
```

Number of duplication in hhid in each file:

```
names(duph) ← fn.0[-1]
duph
```

```
 s10  s11  s12  s13 s14a s14b s15a s15b  s16  s17  s18  s19  s20  s21  s22 s23a
   2   45    3    4    2    3    2    2    3    3    2    3    3    2    2   96
s23b  s24 s02a s02b s02c s03a s03b s04a s04b s05a s05b s06a s06b s07a s08a s08b
   2    3    0    2    2    2    2    2    2    2    2    0    0    2    3    2
 s09
   2
```