Read cleaned GUK files

May 31, 2022
20:54

Seiro Ito

Contents

I have renamed columns, corrected typo's, and set all original column names to lower cases. All variables that I create begin with an upper case letter. All variables beginning with a lower case letter are original variables.

Texts in red indicate relatively major issues in data cleaning. Texts in green show responses to the problem. A Variable Name in red indicates a useful variable that I created.

Several issues discussed with Abu-san on Nov 16, 2017.

- Promissing avenues for impact evaluations: Asset incomes (e.g., milk), schooling (catch up process of large amount arms). There are a few ways to define a treatment status (assignment, elapsed time).
- Saving and repayment info needs to be supplemented with admin data.
- Papers to be written:
  1. Impact evaluation (+ weekly saving and revenue data).
  2. Financial returns.
  3. Relocation impacts.
  4. Repayment pattern and investment choices (Abu-san takes a lead?).
- File maintainer: Abu-san. Anyone who revises the data file should submit to Abu-san and he will update folder in the cloud.

I Read files

ID file, other section files, and roster files are saved in different folders. Correct roster. Check panel structure of each section. Then we examine panel recording status (attrition, membership status, treatment assignment), and attach this to each section files.

List and read files in following folders: ./clean_panel_data_by_section/.

Asset codes are different between rounds, so one cannot use generate.factors = T option of read.dta13 command. I will manually subsitute asset item contents to asset codes. ./only_panel_2_3_4/, ./raw_source_files/P1_Check_20170513, ./raw_source_files/P2_Check_20170513, ./raw_source_files/P3_Check_20170513, ./raw_source_files/P4_Check_20170513.

Add the roster to the list of 1-2-3-4 panel data files (Z) as an element. (Note: At this moment, the list element roster is also a list, not a data.table.) Save all files. Z uses files from panel1234 and roster, Z2 uses files from panel234.

```
saveRDS(Z, paste0(path1234, "data_read_in_a_list_AssetsCodeOnly.rds"))
saveRDS(Z2, paste0(path1234, "data_read_in_a_list_234.rds"))
```

II Id

II.1 Membership status

Check individual panel status: gid is borrower group's unique id, and hhid = gid+membership ID number. member_mid is missing for some but memname is nonNA for all.

```
          memberNA
memnameNA FALSE TRUE
    FALSE  4093 4296
```

In below, I create several variables to show membership-attrition patterns.

Define Mpasted: Tabulate membership pattern across rounds. It is cccc if membership==continued for 4 rds, d if dropped out after rd 1 while dd if observed in rd 1 but dropped out in rd 2 or 3. It does not give timing of attrition. Note: Original membership is defined at each observed rounds.

c continuing members (original members, agreed to participate)

d dropped out members (original members, dropped out by flood, individual rejection, group rejection)

n new members (members of newly added group)

r replacing members for dropped out members (additional members, replacing dropped out members in the original group)

Survey team tried to track an idividual who dropped out at rd 1, so such a person is observed or is lost all the way to the final round.

```
   c    cc   ccc cccc    d    dd   ddd dddd    n    nn   nnn nnnn    r    rr   rrr rrrr
  25    30    84 4612   68    30   201  916    5     4    39 1760    2     4    33  576
```

Create survey round pattern: Spasted.

```
       Spasted
survey    1   12  123 1234  124   13  134
     1  100   28   82 1966    1    6   36
     2    0   28   82 1966    1    0    0
     3    0    0   82 1966    0    6   36
     4    0    0    0 1966    1    0   36
```

Create Mpattern which shows membership and attrition information.

```
xid[, Mpattern := Mpasted]
# if Spasted == 1, observed only in rd1, so Xaaa (a for attrition)
xid[Spasted == 1, Mpattern := paste0(Mpasted, "aaa")]
xid[Spasted == 12, Mpattern := paste0(Mpasted, "aa")]
xid[Spasted == 13, Mpattern := paste0(substr(Mpasted, 1, 1), "a",
        substr(Mpasted, 2, 2), "a")]
xid[Spasted == 123, Mpattern := paste0(Mpasted, "a")]
xid[Spasted == 124, Mpattern := paste0(substr(Mpasted, 1, 2), "a",
        substr(Mpasted, 3, 3))]
xid[Spasted == 134, Mpattern := paste0(substr(Mpasted, 1, 1), "a",
        substr(Mpasted, 2, 3))]
```

Tabulate membership-attrition patten:

```
caaa caca cacc ccaa ccac ccca cccc daaa dada dadd ddaa ddda dddd naaa nann nnaa
  25    8   42   22    3   39 4612   68    2   39   28  162  916    5   12    4
nnna nnnn raaa rara rarr rraa rrra rrrr
  27 1760    2    2   15    2   18  576
```

(Note above number must be adjusted in the following way: Since ccaa is observed only in 2 rds, number of individuals is 22/2 = 11.)

II.2 Treatment assignment

Tabulate Mpattern against reason for current membership status (membership_status).

```
        Mstatus
Mpattern gErosion gRejection iRejection iReplacement newGroup oldMember
    caaa        0          0          0            0        0        25
    caca        0          0          0            0        0         8
    cacc        0          0          0            0        0        42
    ccaa        0          0          0            0        0        22
    ccac        0          0          0            0        0         3
```

ccca	0	0	0	0	0	39
cccc	0	0	3	0	0	4609
daaa	24	22	22	0	0	0
dada	2	0	0	0	0	0
dadd	0	0	39	0	0	0
ddaa	4	8	14	0	0	2
ddda	159	0	3	0	0	0
dddd	0	342	574	0	0	0
naaa	0	0	0	0	5	0
nann	0	0	0	0	12	0
nnaa	0	0	0	0	4	0
nnna	0	0	0	0	27	0
nnnn	0	0	0	0	1760	0
raaa	0	0	0	2	0	0
rara	0	0	0	2	0	0
rarr	0	0	0	15	0	0
rraa	0	0	0	2	0	0
rrra	0	0	0	18	0	0
rrrr	0	0	0	576	0	0

Below is original members.

```
table0(xid[survey == 1 & !grepl("new|Rep", Mstatus), Mstatus])
```

```
  gErosion gRejection iRejection  oldMember
        80         140         159        1221
```

Create Mgroup which identifies continued, newly added group (after flood?), or members replacing rejecters.

gErosion Forced drop outs.

gRejection, iRejection Voluntary drop outs.

There is an anomaly in membership_status: Given that gid==71372 does not reject loans by group, this must be a drop out due to individual rejection rather than Old member that membership_status reports. Correct Mgroup and Mstatus accordingly (but keep original membership_status unchanged).

```
xid[grepl("Dr", membership) & grepl("Old", membership_status),
        .(gid, hhid, year, Mgroup, Mpattern, Mstatus, membership, membership_status,
        creditstatus, missing_followup)]
```

```
     gid    hhid year    Mgroup Mpattern   Mstatus      membership
1: 71372 7137219 2012 continued     ddaa oldMember Drop-out member
2: 71372 7137219 2014 continued     ddaa oldMember Drop-out member
   membership_status creditstatus          missing_followup
1:        Old Member           No 3rd and 4th round missing
2:        Old Member           No 3rd and 4th round missing
```

```
xid[grepl("Dr", membership) & grepl("Old", membership_status),
        c("Mgroup", "Mstatus") := list("drop outs", "iRejection")]
```

```
              Mgroup
Mstatus        continued drop outs forced drop outs new group replacements
  gErosion             0          0             189         0            0
  gRejection           0        372               0         0            0
  iRejection           3        654               0         0            0
  iReplacement         0          0               0         0          615
  newGroup             0          0               0      1808            0
  oldMember         4748          0               0         0            0
```

Create Assign which shows realised assignment (as opposed to original assignment randomization) and drop out status (Mstatus).

```
xid[, AssignOriginal := randomization]
xid[, AssignOriginal := gsub("^con.*", "traditional", AssignOriginal)]
xid[, AssignOriginal := gsub("L.*t$", "large", AssignOriginal)]
xid[, AssignOriginal := gsub("L.*d.$", "large grace", AssignOriginal)]
xid[, AssignOriginal := gsub("^p.*", "cow", AssignOriginal)]

xid[, Assign := AssignOriginal]
xid[grepl("^dr", Mgroup), Assign := "drop outs"]
xid[grepl("^fo", Mgroup), Assign := "forced drop outs"]
```

Tabulate AssignOriginal in the first round. (Note: 220 NAs will be dealt with in the impact estimation file using village level info.)

```
        cow       large large grace traditional        <NA>
        512         472         482         533         220
```

```
              Mgroup
AssignOriginal continued drop outs forced drop outs new group replacements
   cow               308         72                0        60           72
   large             348         12                0       100           12
   large grace       338         22                0       100           22
   traditional       227         53                0       200           53
   <NA>                0        140               80         0            0
```

```
              Mstatus
AssignOriginal gErosion gRejection iRejection iReplacement newGroup oldMember
   cow                0          0         72           72       60        308
   large              0          0         12           12      100        348
   large grace        0          0         22           22      100        338
   traditional        0          0         54           53      200        226
   <NA>              80        140          0            0        0          0
```

Tabulate Mpattern against assignment status (Assign).

```
        Assign
Mpattern traditional large large grace  cow drop outs forced drop outs
    caaa           2    10           4    9         0                0
    caca           0     4           0    4         0                0
    cacc           3     6          21   12         0                0
    ccaa           2     4           6   10         0                0
    ccac           0     0           3    0         0                0
    ccca           3     6          18   12         0                0
    cccc         888  1320        1268 1136         0                0
    daaa           0     0           0    0        44               24
    dada           0     0           0    0         0                2
    dadd           0     0           0    0        39                0
    ddaa           0     0           0    0        24                4
    ddda           0     0           0    0         3              159
    dddd           0     0           0    0       916                0
    naaa           2     1           2    0         0                0
    nann           3     3           3    3         0                0
    nnaa           0     2           2    0         0                0
    nnna          27     0           0    0         0                0
    nnnn         752   388         384  236         0                0
    raaa           0     1           0    1         0                0
    rara           0     0           0    2         0                0
    rarr           0     0           0   15         0                0
```

rraa	0	0	0	2	0	0
rrra	6	0	0	12	0	0
rrrr	204	44	88	240	0	0

Tabulate Mstatus against realiased assignment status (Assign).

	Assign					
Mstatus	traditional	large	large grace	cow	drop outs	forced drop outs
gErosion	0	0	0	0	0	189
gRejection	0	0	0	0	372	0
iRejection	3	0	0	0	654	0
iReplacement	210	45	88	272	0	0
newGroup	784	394	391	239	0	0
oldMember	895	1350	1320	1183	0	0

Missingness was reported with errors but corrected with updated file of 2017-10-18. Tabulate Mpattern against attrition information (missing_followup).

	Mpattern								
missing_followup	caaa	caca	cacc	ccaa	ccac	ccca	cccc	daaa	dada
First follow-up missing	0	0	42	0	0	0	0	0	0
Second follow-up missing	0	0	0	0	3	0	0	0	0
Endline missing	0	0	0	0	0	39	0	0	0
2nd and 4th round missing	0	8	0	0	0	0	0	0	2
3rd and 4th round missing	0	0	0	22	0	0	0	0	0
2nd, 3rd and 4th round missing	25	0	0	0	0	0	0	68	0
None missing	0	0	0	0	0	0	4612	0	0

	Mpattern								
missing_followup	dadd	ddaa	ddda	dddd	naaa	nann	nnaa	nnna	nnnn
First follow-up missing	39	0	0	0	0	12	0	0	0
Second follow-up missing	0	0	0	0	0	0	0	0	0
Endline missing	0	0	162	0	0	0	0	27	0
2nd and 4th round missing	0	0	0	0	0	0	0	0	0
3rd and 4th round missing	0	28	0	0	0	0	4	0	0
2nd, 3rd and 4th round missing	0	0	0	0	5	0	0	0	0
None missing	0	0	0	916	0	0	0	0	1760

	Mpattern					
missing_followup	raaa	rara	rarr	rraa	rrra	rrrr
First follow-up missing	0	0	15	0	0	0
Second follow-up missing	0	0	0	0	0	0
Endline missing	0	0	0	0	18	0
2nd and 4th round missing	0	2	0	0	0	0
3rd and 4th round missing	0	0	0	2	0	0
2nd, 3rd and 4th round missing	2	0	0	0	0	0
None missing	0	0	0	0	0	576

Timing of disbursement.

Create DistDateX to show the timing of intervention in terms of survey. There are 3 disbursements for traditional loans, so DistDate2, DistDate3 are defined only for them. When DistDateX==NA and PurDateX!=NA, use PurDateX to fill NAs in DistDateX.

```
xid[is.na(DistDate1) & !is.na(PurDate1), DistDate1 := PurDate1]
xid[is.na(DistDate2) & !is.na(PurDate2), DistDate2 := PurDate2]
```

Define DisbursedX: T if interview date is after the X-th disbuersement date, F otherwise.

```
xid[DistDate1 > IntDate, Disbursed1 := F]
xid[DistDate1 ≤ IntDate, Disbursed1 := T]
xid[DistDate2 > IntDate, Disbursed2 := F]
xid[DistDate2 ≤ IntDate, Disbursed2 := T]
xid[DistDate3 > IntDate, Disbursed3 := F]
xid[DistDate3 ≤ IntDate, Disbursed3 := T]
```

If DistDate1==NA and creditstatus==No, set Disbursed1=F (except for drop outs). Given that it is only the traditional loan takers which match this pattern, set also Disbursed2, Disbursed3 to F.

```
xid[is.na(DistDate1) & is.na(DistDate2) & is.na(DistDate3) &
      !grepl("dr", Mgroup) & grepl("N", creditstatus),
      c("Disbursed1", "Disbursed2", "Disbursed3") := F]
```

If DistDate1==NA and Mgroup==drop outs or Mstatus=={iRejection, gErosion, gRejection}, set DisbursedX==F.

```
xid[is.na(DistDate1) & (grepl("dr", Mgroup) | grepl("R|E", Mstatus)), Disbursed1 := F]
```

If DistDate1!=NA and IntDate==NA, set subsequent (than DistDate1) rd (survey) of Disbursed1 to T.

```
xid[!is.na(DistDate1) & is.na(IntDate), Disbursed1 := F]
xid[!is.na(DistDate1) & is.na(IntDate), DisRd := survey]
xid[!is.na(DistDate1) & is.na(IntDate) & survey > DisRd, Disbursed1 := T]
xid[, DisRd := NULL]
```

Who are these who did not receive loans but Mgroup is classified as a continuing member?

```
    year DistDate1       Mgroup      Assign Mpattern      Mstatus
 1: 2012      <NA>    continued traditional     cccc    oldMember
 2: 2014      <NA>    continued traditional     cccc    oldMember
 3: 2015      <NA>    continued traditional     cccc    oldMember
 4: 2017      <NA>    continued traditional     cccc    oldMember
 5: 2013      <NA> replacements traditional     rrrr iReplacement
 6: 2014      <NA> replacements traditional     rrrr iReplacement
 7: 2015      <NA> replacements traditional     rrrr iReplacement
 8: 2017      <NA> replacements traditional     rrrr iReplacement
 9: 2013      <NA>    new group traditional     nnnn     newGroup
10: 2014      <NA>    new group traditional     nnnn     newGroup
11: 2015      <NA>    new group traditional     nnnn     newGroup
12: 2017      <NA>    new group traditional     nnnn     newGroup
13: 2013      <NA>    new group traditional     nnna     newGroup
14: 2014      <NA>    new group traditional     nnna     newGroup
15: 2015      <NA>    new group traditional     nnna     newGroup
         membership      membership_status creditstatus Count
 1:       Continued             Old Member           No    26
 2:       Continued             Old Member           No    26
 3:       Continued             Old Member           No    26
 4:       Continued             Old Member           No    26
 5: Replaced member Individual Replacement           No     3
 6: Replaced member Individual Replacement           No     3
 7: Replaced member Individual Replacement           No     3
 8: Replaced member Individual Replacement           No     3
 9:      New member               New Group           No    18
10:      New member               New Group           No    18
11:      New member               New Group           No    18
12:      New member               New Group           No    18
13:      New member               New Group           No     2
14:      New member               New Group           No     2
15:      New member               New Group           No     2
```

hhids of the above.

```
 [1]     7042505     7042507     7042512     7042513     7042518     7065004
 [7]     7065007     7065017     7086111     7086113     7086115     7086116
[13]     7086117     7086118     7086119     7116614     7116615     7137201
[19]     7137209     7137212     8169303     8169305     8169306     8169316
[25]     8169317     8169320  9807065005  9807065009  9807065015  9907065112
[31]  9907065113 99070310710 99070310715 99070311401 99070311404 99070311406
```

```
[37] 99070311409 99070311410 99070311413 99070311414 99070311417 99070311418
[43] 99070311420 99070311503 99070311504 99070311506 99070311510 99070311518
[49] 99070311519
```

[2017-11-14 Abu email] These individuals are loan rejecters yet stay as a member. → Mark as rejecters by creating a variale BorrowerStatus={borrower, pure save}.

```
xid[, BorrowerStatus := "borrower"]
xid[is.na(DistDate1) & is.na(DistDate2) & is.na(DistDate3) & !grepl("dr", Mgroup),
        BorrowerStatus := "pure saver"]
```

ObPattern.

```
0111 1000 1010 1011 1100 1110 1111  Sum
  36  100    6    1   28   82 1966 2219
```

AttritIn.

```
     AttritIn
Tee     2    3    4    9  Sum
  1   100    0    0    0  100
  2     0   56    0    0   56
  3     0    0  258    0  258
  4     0    0    0 7975 7975
  Sum 100   56  258 7975 8389
```

Save xid.

```
saveRDS(xid, paste0(path1234, "ID.rds"))
```

Traditional loans are disbursed 3 times.

```
   DistDate1                    DistDate2
 Min.   :2013-04-16 00:00:00   Min.   :2014-03-23 00:00:00
 1st Qu.:2013-04-22 00:00:00   1st Qu.:2014-03-23 00:00:00
 Median :2013-05-05 00:00:00   Median :2014-03-23 00:00:00
 Mean   :2013-06-30 03:25:18   Mean   :2014-05-29 01:09:23
 3rd Qu.:2013-09-29 00:00:00   3rd Qu.:2014-09-14 00:00:00
 Max.   :2013-11-18 00:00:00   Max.   :2014-09-14 00:00:00
   DistDate3                                 Assign
 Min.   :2015-02-11 00:00:00   traditional     :498
 1st Qu.:2015-02-11 00:00:00   large           :  0
 Median :2015-02-11 00:00:00   large grace     :  0
 Mean   :2015-04-21 18:30:21   cow             :  0
 3rd Qu.:2015-08-11 00:00:00   drop outs       :  0
 Max.   :2015-08-11 00:00:00   forced drop outs:  0
```

Drop outs did not receive loans.

```
      gid           survey       DistDate1      DistDate2      DistDate3
 70319   : 80   Min.   :1.00   Min.   :NA     Min.   :NA     Min.   :NA
 70858   : 80   1st Qu.:1.00   1st Qu.:NA     1st Qu.:NA     1st Qu.:NA
 81483   : 80   Median :2.00   Median :NA     Median :NA     Median :NA
 70317   : 78   Mean   :2.33   Mean   :NA     Mean   :NA     Mean   :NA
 81697   : 77   3rd Qu.:3.00   3rd Qu.:NA     3rd Qu.:NA     3rd Qu.:NA
 70539   : 71   Max.   :4.00   Max.   :NA     Max.   :NA     Max.   :NA
 (Other):749                   NA's   :1215   NA's   :1215   NA's   :1215
              Assign               creditstatus
 traditional     :   0   Yes             :  0
 large           :   0   No              :540
 large grace     :   0   Replaced Member:  0
 cow             :   0   NA's            :675
 drop outs       :1026
 forced drop outs: 189
```

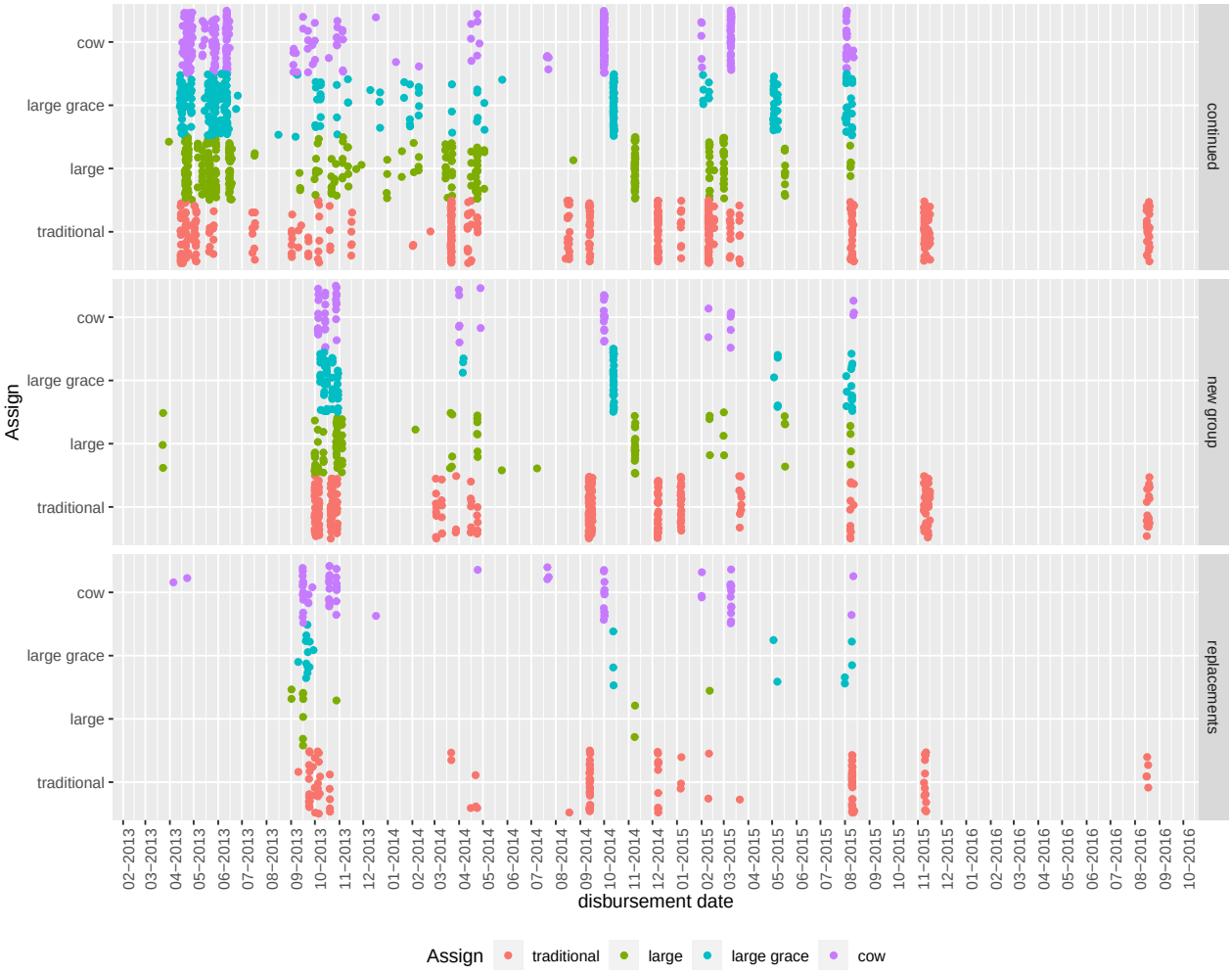

Figure 1 Disbursement timing

Figure 2 Disbursement progress of first loans against interview dates

Plot disbursement timing after excluding rejecters and drop-out members (Figure 1). Note that continuing members are the original members.

Plot disbursement status against interview dates (Figure 2) and disbursement dates (Figure 5). (After correcting some typos before date conversion.) We plot first loan disbursment against against disbursement dates (Figure 6), and calendar year (Figure 3).

```
         Assign       Mgroup      Mstatus Count
1:        large replacements iReplacement     4
2: traditional replacements iReplacement    14
```

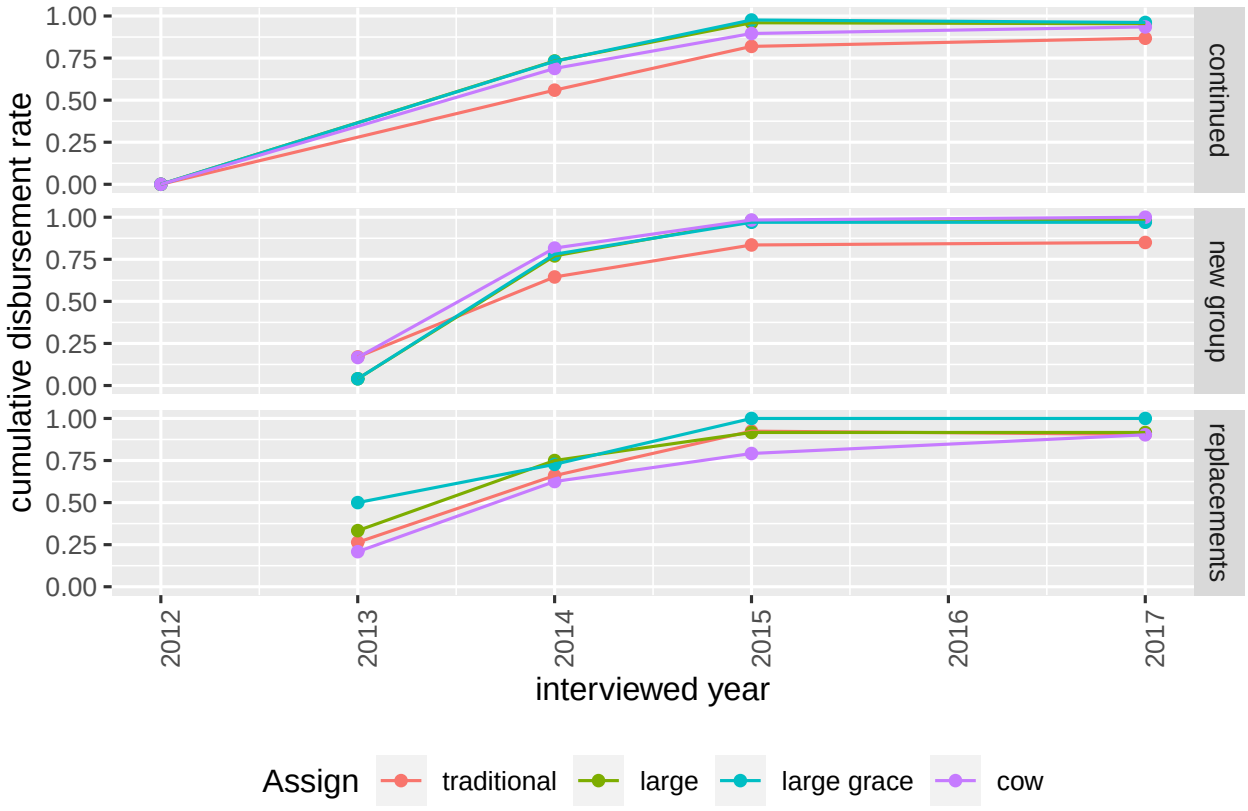

Figure 3 Disbursement progress of first loans against calendar year

```
 3:  traditional replacements iReplacement    12
 4:          cow replacements iReplacement    15
 5:          cow replacements iReplacement    14
 6:          cow replacements iReplacement    13
 7:  large grace replacements iReplacement    11
 8:  traditional     new group     newGroup    34
 9:  traditional     new group     newGroup    31
10:        large     new group     newGroup     4
11:          cow     new group     newGroup    10
12:  large grace     new group     newGroup     4
13:  large grace     new group     newGroup     3
```

There are 44 cases which received treatment at the first round of survey among Mgroup==replacements, and 52 cases for Mgroup==new group. These do not have baseline and needs to be dropped from analysis. The progress is shown in Figure 4.

[2017-11-17 Abu discussion] These disbursement dates are wrong and need to be replaced with information in administrative records. → Abu will send the admin files. Not received as of 2019 Feb.

III Correct sections

All files are corrected. Only roster is merged with xid at this point. All the other files are merged with xid in the next section.

III.1 Roster (xid merged)

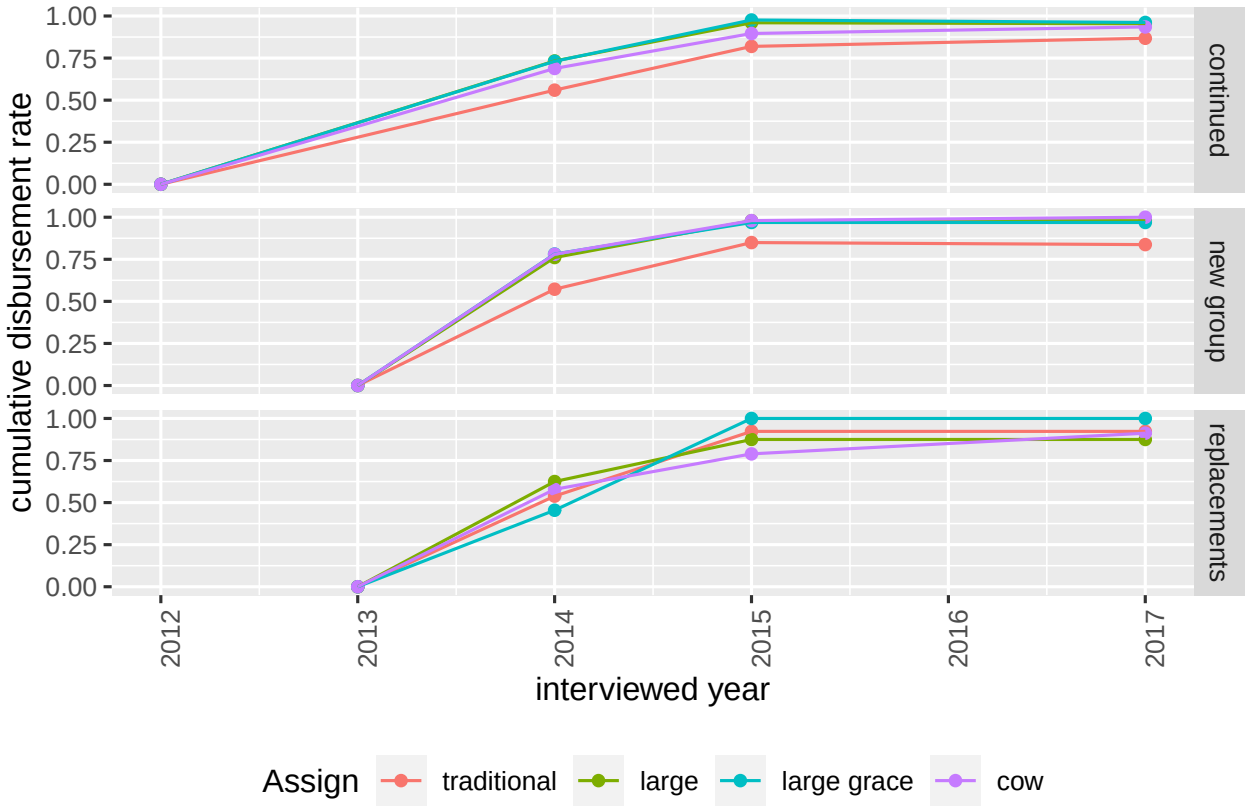

Figure 4 Disbursement progress of first loans against calendar year after dropping obs without baseline

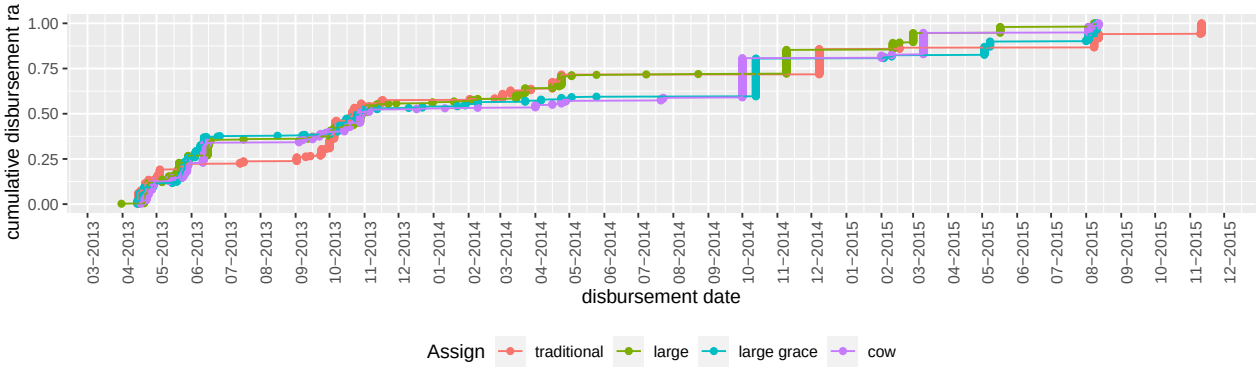

Figure 5 Disbursement progress of first loans against disbursement dates

```
Z3 ← Z[[grep("roster", names(Z))]]
NAhhid ← lapply(Z3, function(x) nrow(x[is.na(hhid), ]))
```

NAs in hhid in roster. Folder s1.1, number of NAs 5. (Note: At this moment, roster is saved as a list, not a data.table.) NAs in mid in roster. Folder s1_2_p2, section_1_houdehold_composition_2, s1.2, number of NAs 1773, 1843, 1966. These look like redundant entries so we can drop with mid==NA. Membership information current is not recorded in 2012, however, most but 10 cases are reportedly staying in HH. So I create current in 2012 by using stay. Other corrections include: Drop duplicates: hhid==7010112 & mid==5 and hhid==7053905 & mid==3 & current == 3, correct mid: hhid==7020605 & mid == 3 & year == 2015: mid 3→4. Filled in NAs in sex if other rds are available. Not sure where Shaha Alom came from in HH 98081710316 in 2017. (Jahanara?)

```
Z3new[is.na(current) & year == 2012 & grepl("y", stay), current := "member"]
Z3new[is.na(current) & year == 2012 & grepl("n", stay), current := "not-member"]
```

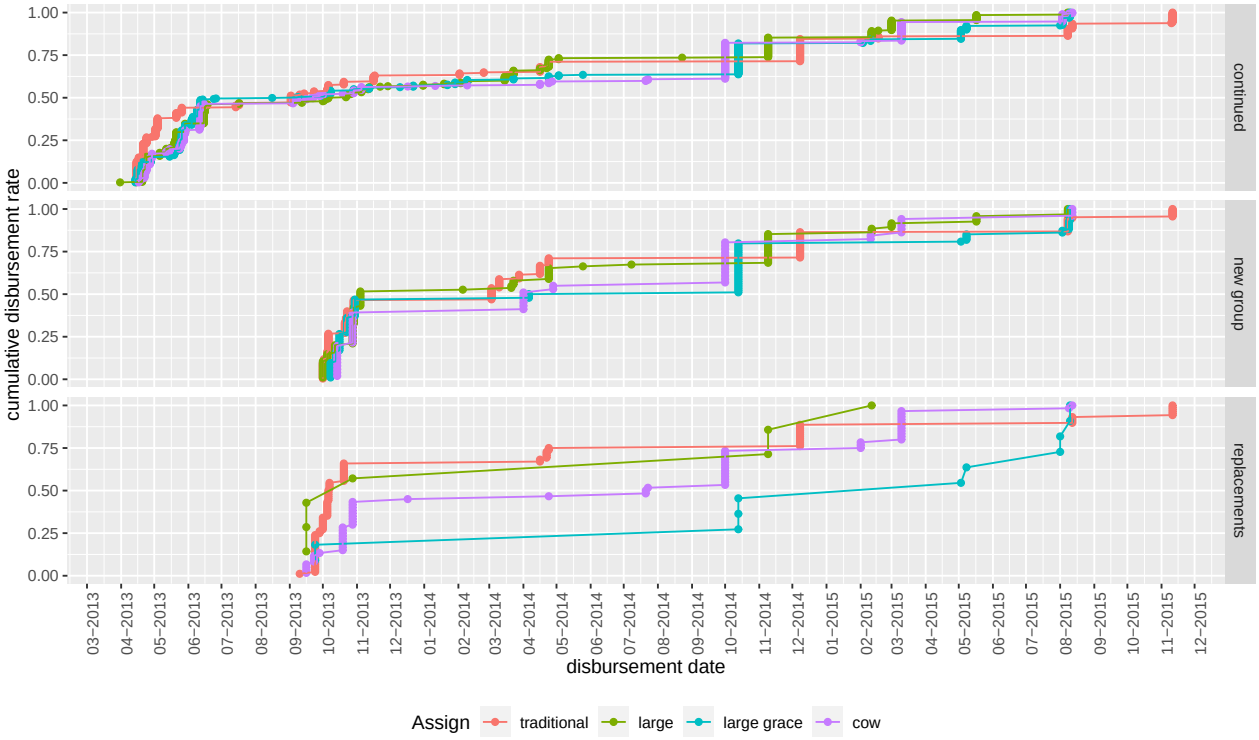

Figure 6 Disbursement progress of first loans by membership status

```
table0(Z3new[, .(year, current)])
```

```
      current
year      1    2    3 member new member not-member <NA>
  2012    0    0    0   9264          0         10    0
  2014    0    0    0   8318        348        352    0
  2015 8364  347  281      0          0          0    1
  2017    2    0   38   8204          0         70    0
```

```
       hhid mid year memname      rel_hhh                       edu stay current
 1: 7010112   1 2012    bablu         head     never been to school  yes  member
 2: 7010112   1 2014    bablu         head                       99  yes  member
 3: 7010112   1 2015    bablu         head                       99  yes       1
 4: 7010112   1 2017    bablu         head                       99  yes  member
 5: 7010112   2 2012   farida       spouse     never been to school  yes  member
 6: 7010112   2 2014   farija husband/wife                       99  yes  member
 7: 7010112   2 2015   farida husband/wife                       99  yes       1
 8: 7010112   2 2017   farida husband/wife                       99  yes  member
 9: 7010112   3 2012   dulifa son/daughter class 2/finished class 2  yes  member
10: 7010112   3 2014   julufa son/daughter                        3  yes  member
11: 7010112   3 2015   julufa son/daughter                        4  yes       1
12: 7010112   3 2017   julufa son/daughter                        6  yes  member
13: 7010112   4 2012    afika son/daughter class 1/finished class 1  yes  member
14: 7010112   4 2014    afika son/daughter                        2  yes  member
15: 7010112   4 2015    afika son/daughter                        3  yes       1
16: 7010112   4 2017    afika son/daughter                        5  yes  member
17: 7010112   5 2015    arifa son/daughter                        1  yes       2
      marital
 1:   married
 2:         2
 3:         2
 4:   married
 5:   married
```

```
 6:         2
 7:         2
 8:   married
 9: unmarried
10:         1
11:         1
12: unmarried
13: unmarried
14:         1
15:         1
16: unmarried
17: unmarried
```

```
       hhid mid year      memname      rel_hhh                   edu stay
 1: 7020605   1 2012       aminul         head never been to school  yes
 2: 7020605   1 2014       aminul         head                   99  yes
 3: 7020605   1 2015       aminul         head                   99   no
 4: 7020605   1 2017       aminul         head                   99  yes
 5: 7020605   2 2012 sona khatun        spouse never been to school  yes
 6: 7020605   2 2014 sona khatun husband/wife                   99  yes
 7: 7020605   2 2015 sona khatun husband/wife                   99  yes
 8: 7020605   2 2017 sona khatun husband/wife                   99  yes
 9: 7020605   3 2012    sona mia son/daughter pre-school going age  yes
10: 7020605   3 2014     sonamia son/daughter                <NA> <NA>
11: 7020605   3 2015  amir hamza son/daughter                   68  yes
12: 7020605   4 2017  amir hamza son/daughter                   68  yes
       current   marital
 1:     member   married
 2:     member         2
 3:          1         2
 4:     member   married
 5:     member   married
 6:     member         2
 7:          1         2
 8:     member   married
 9:     member unmarried
10: not-member      <NA>
11:          3 unmarried
12:          1 unmarried
```

```
           hhid mid     memname age_1 year stay current   marital
 1: 98081710316   1 pero mondol    85 2012  yes  member   widowed
 2: 98081710316   1  pirumondol    NA 2015  yes       1         3
 3: 98081710316   1  pirumondol    NA 2017  yes  member   widowed
 4: 98081710316   2      safura    50 2012  yes  member   married
 5: 98081710316   2      sofura    NA 2015 <NA>       2      <NA>
 6: 98081710316   3    jahanara    45 2012  yes  member   married
 7: 98081710316   3    jahanara    NA 2015  yes       1         2
 8: 98081710316   4      afroza     7 2012  yes  member unmarried
 9: 98081710316   4      afruja    NA 2015  yes       1         2
10: 98081710316   4      afruja    NA 2017  yes  member   married
11: 98081710316   5        siam    NA 2017  yes       3 unmarried
12: 98081710316   6  shaha alom    NA 2017  yes       1   married
```

```
, , year = 2012

      current
stay   member new member not-member <NA>
  no        0          0         10    0
  yes    9264          0          0    0
```

```
  <NA>       0           0           0     0

, , year = 2014

      current
stay    member new member not-member <NA>
  no        43           3           0     0
  yes     8275         345           1     0
  <NA>       0           0         351     0

, , year = 2015

      current
stay    member new member not-member <NA>
  no         0           0           0    52
  yes        0           0           0 8593
  <NA>       0           0           0   345

, , year = 2017

      current
stay    member new member not-member <NA>
  no        27           0           0     0
  yes     8179           0           0    38
  <NA>       0           0          70     0
```

Add survey using ID (xid) file.

```
xid ← readRDS(paste0(path1234, "ID.rds"))
xid2 ← unique(xid[, .(gid, hhid, povertystatus, year, survey, memname,
        creditstatus, Mpattern, Mgroup, Mstatus, Assign, AssignRegression,
        ObPattern, AttritIn,
        IntDate, DistDate1, DistDate2, DistDate3, Disbursed1, Disbursed2, Disbursed3)])
xid3 ← unique(xid2[, .(hhid, year, survey)])
setnames(xid3, "year", "YearFromIdFile")
setkey(xid3, hhid, survey)
setkey(Z3new, hhid, survey)
Z3new2 ← xid3[Z3new]
setkey(Z3new2, hhid, mid, YearFromIdFile, year, survey)
Z3new2[, .(hhid, mid, YearFromIdFile, year, survey, age)]
```

```
             hhid mid YearFromIdFile year survey age
    1:    7010101   1            2012 2012      1  40
    2:    7010101   1            2014 2014      2  NA
    3:    7010101   1            2015 2015      3  NA
    4:    7010101   1            2017 2017      4  NA
    5:    7010101   2            2012 2012      1  35
   ---
35592: 99081912420   3            2017 2017      4  NA
35593: 99081912420   4            2013 2012      1   7
35594: 99081912420   4            2014 2014      2  NA
35595: 99081912420   4            2015 2015      3  NA
35596: 99081912420   4            2017 2017      4  NA
```

```
Z3new2[, year := YearFromIdFile]
```

```
[1] 8 duplicated entries.
```

3 of duplicated entries in roster are errors (new borns), others are use of attrited member mid (e.g., mid = 2 left HH and mid = 3 uses mid 2 from that time on). There may be other cases of nonunique assignment of mid, but most of the times it will be picked up by an ID duplication check.

```
dupZ3[duplicated.num == 3 | duplicated.num == 4 | duplicated.num == 7, ]
```

```
   duplicated not.duplicated duplicated.num    hhid mid year AgeComputed
1:      FALSE          FALSE              3 7042213   6 2017          NA
2:      FALSE          FALSE              3 7042213   6 2017          NA
3:      FALSE          FALSE              4 7042417   5 2015          NA
4:      FALSE          FALSE              4 7042417   5 2015          NA
5:      FALSE          FALSE              7 7096319   5 2017          NA
6:      FALSE          FALSE              7 7096319   5 2017          NA
   memname      rel_hhh   marital             literacy edu primary secondary
1:   sathi son/daughter unmarried can't read and write  99   child      <NA>
2:   sathi son/daughter unmarried can't read and write  99   child      <NA>
3: ibrahim son/daughter         1 can't read and write  68   child      <NA>
4: ibrahim son/daughter unmarried can't read and write  68   child      <NA>
5:   munni son/daughter unmarried can't read and write  68   child      <NA>
6:   munni son/daughter unmarried can't read and write  68   child      <NA>
         uid age month current          reasons reason FirstObs FirstObs2
1:        NA  NA    NA  member             <NA>   <NA>        0         0
2:        NA   2     1    <NA> new born/adopted   <NA>        0         0
3: 704241705  NA    NA    <NA>             <NA>   <NA>        0         0
4: 704241705   1     6    <NA>                1  flood        0         0
5:        NA  NA    NA  member             <NA>   <NA>        1         1
6:        NA   0     9    <NA> new born/adopted   <NA>        0         0
```

Drop the onew with age==NA.

```
Z3new2 ← Z3new2[
        !(hhid == 7042213 & mid == 6 & year == 2017 & is.na(age)) &
        !(hhid == 7042417 & mid == 5 & year == 2015 & is.na(age)) &
        !(hhid == 7096319 & mid == 5 & year == 2017 & is.na(age)), ]
Z3new2[hhid == 7010112, .(hhid, mid, memname, year, AgeComputed, age)]
```

```
       hhid mid memname year AgeComputed age
 1: 7010112   1    bablu 2012          45  45
 2: 7010112   1    bablu 2014          47  NA
 3: 7010112   1    bablu 2015          48  NA
 4: 7010112   1    bablu 2017          50  NA
 5: 7010112   2   farida 2012          32  32
 6: 7010112   2   farija 2014          34  NA
 7: 7010112   2   farida 2015          35  NA
 8: 7010112   2   farida 2017          37  NA
 9: 7010112   3   dulifa 2012          10  10
10: 7010112   3   julufa 2014          12  NA
11: 7010112   3   julufa 2017          15  NA
12: 7010112   4    afika 2012           5   8
13: 7010112   4    afika 2014           7  NA
14: 7010112   4   julufa 2015           8  NA
15: 7010112   4    afika 2015           8  NA
16: 7010112   4    afika 2017          10  NA
```

```
Z3new2[hhid == 7010112 & year == 2015 & grepl("jul", memname), mid := 3]
Z3new2[hhid == 7020811, .(hhid, mid, memname, year, AgeComputed, age)]
```

```
       hhid mid  memname year AgeComputed age
 1: 7020811   1   mounal 2012          35  35
 2: 7020811   1   moynal 2014          37  NA
 3: 7020811   1   mounal 2015          38  NA
 4: 7020811   1   mounal 2017          40  NA
 5: 7020811   2   riaton 2012          30  30
 6: 7020811   2   riaton 2014          32  NA
 7: 7020811   2   riaton 2015          33  NA
```

```
 8: 7020811   2   riaton 2017          35  NA
 9: 7020811   3  monuara 2012          12  12
10: 7020811   3  monoara 2014          14  NA
11: 7020811   3 monowara 2015          15  NA
12: 7020811   4    iasin 2012           8   8
13: 7020811   4    iasin 2014          10  NA
14: 7020811   4   eyasin 2015          11  NA
15: 7020811   4   eyasin 2017          13  NA
16: 7020811   5   masuma 2014          NA   0
17: 7020811   5   masuma 2015          NA  NA
18: 7020811   5     runa 2015          NA   0
19: 7020811   6     runa 2017          NA  NA
```

```
Z3new2[hhid == 7020811 & year == 2015 & grepl("run", memname), mid := 6]
Z3new2[hhid == 7054105, .(hhid, mid, memname, year, AgeComputed, age)]
```

```
       hhid mid memname year AgeComputed age
 1: 7054105   1  samsul 2012          60  60
 2: 7054105   1  samsul 2014          62  NA
 3: 7054105   1  rahima 2015          63  NA
 4: 7054105   2  rahima 2012          50  50
 5: 7054105   2  rahima 2014          52  NA
 6: 7054105   2    modu 2015          53  NA
 7: 7054105   2  rahima 2017          55  NA
 8: 7054105   3    modu 2012           7   7
 9: 7054105   3    modu 2014           9  NA
10: 7054105   3  rahman 2014           9  38
11: 7054105   3  rahman 2015          10  NA
12: 7054105   3    modu 2017          12  NA
13: 7054105   4  jorina 2014          NA  30
14: 7054105   4  jorina 2015          NA  NA
15: 7054105   4  rahman 2017          NA  NA
16: 7054105   5  soneka 2014          NA   9
17: 7054105   5  soneka 2015          NA  NA
18: 7054105   5  jorina 2017          NA  NA
19: 7054105   6  jesmin 2014          NA   7
20: 7054105   6  jesmin 2015          NA  NA
21: 7054105   6  soneka 2017          NA  NA
22: 7054105   7  jesmin 2017          NA  NA
       hhid mid memname year AgeComputed age
```

```
Z3new2[hhid == 7054105 & grepl("rahima", memname), mid := 2]
Z3new2[hhid == 7054105 & grepl("modu$", memname), mid := 3]
Z3new2[hhid == 7054105 & grepl("rahman$", memname), mid := 4]
Z3new2[hhid == 7054105 & grepl("jorina$", memname), mid := 5]
Z3new2[hhid == 7054105 & grepl("soneka$", memname), mid := 6]
Z3new2[hhid == 7054105 & grepl("jesmin$", memname), mid := 7]
Z3new2[hhid == 7096319, .(hhid, mid, memname, year, AgeComputed, age)]
```

```
       hhid mid memname year AgeComputed age
 1: 7096319   1 morjina 2012          50  50
 2: 7096319   1 morjina 2014          52  NA
 3: 7096319   1 morjina 2015          53  NA
 4: 7096319   1 morjina 2017          55  NA
 5: 7096319   2   mojid 2015          NA  35
 6: 7096319   2   mojid 2017          NA  NA
 7: 7096319   3   sujon 2015          NA   8
 8: 7096319   3  sondha 2015          NA  30
 9: 7096319   3   sujon 2017          NA  NA
10: 7096319   4   munni 2015          NA   4
```

```
11: 7096319   4  sondha 2017          NA  NA
12: 7096319   5   munni 2017          NA   0
```

```
Z3new2[hhid == 7096319 & grepl("sujon", memname), mid := 3]
Z3new2[hhid == 7096319 & grepl("sondha", memname), mid := 4]
Z3new2[hhid == 7096319 & grepl("munni", memname), mid := 5]
Z3new2[hhid == 7116604, .(hhid, mid, memname, year, AgeComputed, age)]
```

```
      hhid mid memname year AgeComputed age
1: 7116604   1   achan 2012          50  50
2: 7116604   1   achan 2014          52  NA
3: 7116604   1   aqhan 2015          53  NA
4: 7116604   2   sukni 2012          45  45
5: 7116604   2   sukni 2014          47  NA
6: 7116604   2   sukni 2015          48  NA
7: 7116604   2   sukne 2017          50  NA
8: 7116604   2  jesmin 2017          50  NA
```

```
Z3new2[hhid == 7116604 & grepl("jesm", memname), mid := 3]
dupZ3 ← duplicated.rows(Z3new2, index = c("hhid", "mid", "year", "AgeComputed"),
        returnOnlyDuplicated = T, returnOnlyDistinctCols = T)
```

```
[1] No duplicated entry.
```

Recalculate age. Use all available age information to fill NAs. First, identify newborns who only has months recorded. For the below, there is no clue.

```
      hhid mid memname year en Age age age_1 age_2
1: 7020910   5  sirina 2017  1  NA  NA    NA    NA
2: 7042710   5  afrina 2017  1  NA  NA    NA    NA
3: 7065019   4   panna 2014  1  NA  NA    NA    NA
4: 7065019   4   panna 2015  2  NA  NA    NA    NA
5: 7065019   4   panna 2017  3  NA  NA    NA    NA
6: 7096320   7 mominul 2017  1  NA  NA    NA    NA
7: 7116604   3  jesmin 2017  1  NA  NA    NA    NA
8: 7137316   4  shihab 2017  1  NA  NA    NA    NA
```

For others, copy age from first observed rd to other rds. There are 671 substitution can be made.

```
        AgeComputedNonNA
agenonNA FALSE  TRUE
   FALSE     5     1
   TRUE    671  9255
```

```
      hhid mid memname year en age AgeComputed
1: 7020910   5  sirina 2017  1  NA          NA
2: 7042710   5  afrina 2017  1  NA          NA
3: 7065019   4   panna 2014  1  NA          NA
4: 7065019   4   panna 2015  2  NA          NA
5: 7065019   4   panna 2017  3  NA          NA
6: 7096320   7 mominul 2017  1  NA          NA
7: 7137316   4  shihab 2017  1  NA          NA
```

There are 7 cases of Age==NA remaining. Created several head level variables.

```
Z3new2[grepl("^he", rel_hhh) & grepl("can r.*te", literacy), HeadLiteracy := T]
Z3new2[, c("HeadLiteracy", "HeadAge") :=
        list(HeadLiteracy[grepl("head", rel_hhh)][1], Age[grepl("head", rel_hhh)][1]),
        by = hhid]
Z3new2[, HHsize := .N, by = list(hhid, year)]
```

Save back in data.

```
Z[[grep("roster", names(Z))]] ← Z3new2
# reorder
Z ← Z[c("roster", names(Z)[!grepl("ros", names(Z))])]
fnd ← c("s1 (roster)", fnd)
```

III.2 Incomes

III.2.1 Farm incomes

Save back in data.

```
Z[[grep("farm", names(Z))]] ← xf
Z[[grep("inp", names(Z))]] ← xio
```

III.2.2 Labour incomes

There is a decrease in cases reporting no labour income at 2014. This is due to omission of non-working members in rd 2 onwards.

```
                        year
positive.labour.income 2012 2013 2014 2015 2017
                 FALSE 4928 1885  247  222  190
                 TRUE  1797  652 2259 2352 2272
```

HH-mids reporting no income in rd1 are not reporting anything in later rounds. (Show only first 2 HHs.)

```
       hhid mid year           code_1 totalincome
 1: 7010102   1 2012 Agri Wage Labor        26000
 2: 7010102   1 2014 Agri Wage Labor        42000
 3: 7010102   1 2015 Agri Wage Labor        48600
 4: 7010102   1 2017 Agri Wage Labor        41200
 5: 7010102   2 2012             <NA>            0
 6: 7010102   3 2012 Agri Wage Labor        50000
 7: 7010102   3 2014 Agri Wage Labor        12000
 8: 7010102   3 2015 Agri Wage Labor        69250
 9: 7010102   3 2017 Agri Wage Labor        41800
10: 7010102   4 2012             <NA>            0
11: 7010102   5 2012             <NA>            0
12: 7010103   1 2012 Agri Wage Labor        48000
13: 7010103   2 2012             <NA>            0
14: 7010103   3 2012             <NA>            0
15: 7010103   4 2012             <NA>            0
```

Save back in data.

```
Z[[grep("labo", names(Z))]] ← x12
```

III.3 Assets

HH assets has item coverage that varies across rounds. Importantly, land holding is not covered in round 1. Using purchased_in_last_1_year in HH assets, we recreate round 1 holding information and create variables AmountFilled which includes imputed land holding of round 1, and an imputation indicator Added. We find there is little variation across rounds.

Land holding is also stored in contract and ownership (in all rounds). There are very few records of land that are leased in or out, so contract and ownership has little information on land.

Coverage of other household asset items differ by rounds. We define NLHAssetAmount which is based only on non-livestock assets that are observed in all rounds. NLBroadHAssetAmount is based on all non-livestock asset items.

$$\text{AssetAmount} = \text{NLHAssetAmount} + \text{TotalImputedValue} + \text{PAssetAmount},$$
$$\text{BroadAssetAmount} = \text{NLBroadHAssetAmount} + \text{TotalImputedValue} + \text{PAssetAmount}.$$

TotalImputedValue is livestock holding values. Median unit values are used as prices for imputation.

Coding changes by round so one cannot apply the same code-contents correspondence to all rounds (which read.dta13 function does that caused erroneous reading in asset holding. Corrected). Varying code-contents correspondence also applies to productive assets.

III.3.1 Household assets

Household asset contents.

	survey	type	medianAmount	meanAmount	num
1:	2	agricultural land	19000	39961.482	110
2:	3	agricultural land	44000	66907.974	117
3:	4	agricultural land	66000	105671.642	67
4:	2	almirah/cabinet	1000	1183.194	536
5:	3	almirah/cabinet	1600	1900.733	1481
6:	4	almirah/cabinet	1800	2086.175	1856
7:	1	bicycle	2100	2172.308	65
8:	2	bicycle	3000	3052.439	82
9:	3	bicycle	2000	2612.229	131
10:	4	bicycle	2000	1978.105	153
11:	1	cassette player	350	350.000	2
12:	2	cassette player	2400	2400.000	2
13:	3	cassette player	2600	8733.333	3
14:	4	cassette player	2200	2200.000	2
15:	1	electric fan	1500	1500.000	1
16:	2	electric fan	800	914.545	22
17:	3	electric fan	1000	1171.190	42
18:	4	electric fan	1000	1190.244	41
19:	2	fallow/submerged land	2000	9991.176	34
20:	3	fallow/submerged land	8000	33673.913	23
21:	4	fallow/submerged land	33000	53470.588	34
22:	2	jewelry	1000	1105.556	18
23:	3	jewelry	1200	2676.611	776
24:	4	jewelry	1000	2241.718	1036
25:	1	mobile phone	1000	1285.556	90
26:	2	mobile phone	1200	1172.810	805
27:	3	mobile phone	1000	1018.080	1324
28:	4	mobile phone	800	940.007	1375
29:	2	motorcycle/scooter	2000	38960.000	5
30:	3	motorcycle/scooter	35000	45966.667	6
31:	4	motorcycle/scooter	45000	68333.333	3
32:	2	others	1600	1600.000	2
33:	3	others	2000	6517.647	34
34:	4	others	1100	1266.667	6
35:	1	radio/tv	500	1509.091	11
36:	2	radio/tv	3000	3697.368	38
37:	3	radio/tv	2500	3112.195	41
38:	4	radio/tv	2000	2953.846	39
39:	2	residential land	6000	11783.677	922
40:	3	residential land	16000	25277.937	979
41:	4	residential land	20000	28825.970	928
42:	1	rickshaw/van	3750	3725.000	20
43:	2	rickshaw/van	5000	5633.333	18

```
44:      3          rickshaw/van          4000   8361.765   34
45:      4          rickshaw/van          5000   8260.938   32
46:      1        sewing machine          1200   1710.526   19
47:      2        sewing machine          2000   2578.261   23
48:      3        sewing machine          3900   3812.500   16
49:      4        sewing machine          3500   2863.636   11
50:      1                 solar          8000  11333.333    3
51:      2                 solar          4900   4900.000    2
52:      3                 solar         16000  16375.000    8
53:      4                 solar         17000  16960.000   25
54:      1              tubewell          1200   1313.898 1180
55:      2              tubewell          1400   1473.455 1567
56:      3              tubewell          1200   1264.724 1831
57:      4              tubewell          1000   1085.163 1838
58:      2               vcr/vcp          2000   2000.000    1
59:      3               vcr/vcp          2450 133583.333    6
60:      4               vcr/vcp          1200   1200.000    1
61:      1            wall clock           200    176.667    9
62:      2            wall clock           300    616.667   21
63:      3            wall clock           200    567.000   10
64:      4            wall clock            90    157.727   22
65:      1           wrist watch           200    608.000   15
66:      2           wrist watch           300    760.312   32
67:      3           wrist watch           200   1104.762   21
68:      4           wrist watch           150    421.176   17
    survey                  type medianAmount meanAmount  num
```

Mean assets for household assets.

```
   survey meanNLHA medianNLHA  stdNLHA medianNumNLHA medianNumNetNLHA
1:      1  1397.75       1200   857.66             1                1
2:      2 15716.41       7000 43601.22             3                3
3:      3 28556.84      13400 48482.55             4                3
4:      4 29830.02      15150 50389.58             4                0
```

Some items (agricultural land, almirah/cabinet, fallow/submerged land, jewelry, motorcycle/scooter, others, residential land, vcr/vcp are not recorded in baseline) are observed from round 2.

```
      type
survey agricultural land almirah/cabinet bicycle cassette player electric fan
     1                 0               0      14                0            0
     2                37             187      26                0            5
     3                37             525      46                0           16
     4                22             662      46                1           15
      type
survey fallow/submerged land jewelry mobile phone motorcycle/scooter others
     1                     0       0           36                   0      0
     2                    15       7          258                   2      1
     3                    11     287          443                   0     14
     4                    14     364          455                   1      2
      type
survey radio/tv residential land rickshaw/van sewing machine solar tubewell
     1        5                0           11              10     1       358
     2       14              330           10               8     0       566
     3       16              345           20               8     3       641
     4       13              323           19               6    12       640
      type
survey vcr/vcp wall clock wrist watch
     1       0          5          10
     2       0          7           8
     3       1          3           7
     4       0          5           2
```

Items observed in all rounds are below:

```
 [1] "tubewell"        "mobile phone"    "bicycle"         "wrist watch"
 [5] "sewing machine"  "rickshaw/van"    "wall clock"      "radio/tv"
 [9] "solar"           "electric fan"    "cassette player"
```

Number of households with anomalous asset entries (decrease in non-land household asset values greater than 25000):

```
          hhid year              type amount NLHAssetAmount
 1:    7096217 2012           bicycle   1000           1000
 2:    7096217 2014      mobile phone   2400           6900
 3:    7096217 2014           bicycle   2500           6900
 4:    7096217 2014          tubewell   2000           6900
 5:    7096217 2015          tubewell  15000          39000
 6:    7096217 2015           jewelry   1000          39000
 7:    7096217 2015  cassette player  22000          39000
 8:    7096217 2015      mobile phone   2000          39000
 9:    7096217 2017           jewelry   4000           3200
10:    7096217 2017  almirah/cabinet   2000           3200
11:    7096217 2017          tubewell   1200           3200
12:    7096217 2017      mobile phone   2000           3200
13:    8169717 2012          tubewell   1500           1500
14:    8169717 2014          tubewell   1600           1600
15:    8169717 2014 residential land   6000           1600
16:    8169717 2015          tubewell   1200          82600
17:    8169717 2015  almirah/cabinet   2600          82600
18:    8169717 2015           jewelry    400          82600
19:    8169717 2015      mobile phone   1400          82600
20:    8169717 2015 residential land  36000          82600
21:    8169717 2015      rickshaw/van  80000          82600
22:    8169717 2017          tubewell    400           3300
23:    8169717 2017  almirah/cabinet   2500           3300
24:    8169717 2017           jewelry    600           3300
25:    8169717 2017 residential land  40000           3300
26:    8169717 2017           bicycle   2500           3300
27:    8169717 2017      mobile phone    400           3300
28: 9908147515 2013           bicycle   3000           3000
29: 9908147515 2014          tubewell   1400           6800
30: 9908147515 2014      mobile phone    900           6800
31: 9908147515 2014 residential land   5000           6800
32: 9908147515 2014           bicycle   4500           6800
33: 9908147515 2015      mobile phone    900          44302
34: 9908147515 2015          tubewell   1400          44302
35: 9908147515 2015 residential land  25000          44302
36: 9908147515 2015           jewelry   1200          44302
37: 9908147515 2015           bicycle  42002          44302
38: 9908147515 2015  almirah/cabinet   1000          44302
39: 9908147515 2017 residential land  30000           4900
40: 9908147515 2017          tubewell   1000           4900
41: 9908147515 2017           bicycle   3200           4900
42: 9908147515 2017      mobile phone    700           4900
43: 9908147515 2017  almirah/cabinet    800           4900
44: 9908147515 2017           jewelry    800           4900
          hhid year              type amount NLHAssetAmount
```

For 9908147515, the anomalous decrease is due to bicycle bought at 42002 in 2015, which may be 4200. Correct it. All other HHs are cassette player and rickshaw/ban, which may be possible that they sold off.

Define:

NLHAssets tubewell, mobile phone, bicycle, wrist watch, sewing machine, rickshaw/van, wall clock, radio/tv, solar, electric fan, cassette player

BroadNLHAssets Use all household asset entries.

```
   survey       HA      BHA   nHA  nBHA
1:      1 1313.20   1313.2   450   450
2:      2 2545.14 16013.8 1443 1480
3:      3 2973.93 27621.5 2374 2423
4:      4 2781.11 27736.6 2553 2602
```

Check HHs with anomalous asset values (changes in naorrow net asset values < -50000). There are members who report sharp decline in net assets.

da50K Diff in HH assets greater than 50K. This is mostly due to radio and casette player entries. There are 1 households among o800==1L whose changes in net total asset < -50000 and have assets of values greater than 50000.

This is rickshaw/van.

```
      hhid survey          type amount     H      BH
1: 8169717      3 rickshaw/van  80000 82600 121600
```

```
for (h in da50K)
  print(xha[hhid == h, .(hhid, t=survey, type, amount,
    #NH=NarrowNLHAssetAmount,
    H=NLHAssetAmount, BH=BroadNLHAssetAmount)])
```

```
       hhid t             type amount     H      BH
 1: 8169717 1         tubewell   1500  1500   1500
 2: 8169717 2         tubewell   1600  1600   7600
 3: 8169717 2 residential land   6000  1600   7600
 4: 8169717 3         tubewell   1200 82600 121600
 5: 8169717 3  almirah/cabinet   2600 82600 121600
 6: 8169717 3          jewelry    400 82600 121600
 7: 8169717 3     mobile phone   1400 82600 121600
 8: 8169717 3 residential land  36000 82600 121600
 9: 8169717 3     rickshaw/van  80000 82600 121600
10: 8169717 4         tubewell    400  3300  46400
11: 8169717 4  almirah/cabinet   2500  3300  46400
12: 8169717 4          jewelry    600  3300  46400
13: 8169717 4 residential land  40000  3300  46400
14: 8169717 4          bicycle   2500  3300  46400
15: 8169717 4     mobile phone    400  3300  46400
```

```
#for (h in dna10K[!(dna10K %in% da50K)])
#  print(xha[hhid == h, .(hhid, t=survey, type, amount,
#    NH=NarrowNLHAssetAmount, H=NLHAssetAmount, BH=BroadNLHAssetAmount)])
```

```
for (h in rda50K)
  print(xha[hhid == h, .(hhid, t=survey, type, amount, RevAmount,
    RNH=RNarrowNLHAssetAmount, RH=RNLHAssetAmount, BH=RBroadNLHAssetAmount)])
for (h in rdna10K[!(rdna10K %in% rda50K)])
  print(xha[hhid == h, .(hhid, t=survey, type, amount, RevAmount,
    RNH=RNarrowNLHAssetAmount, NH=NarrowNLHAssetAmount,
    RH=RNLHAssetAmount, H=NLHAssetAmount)])
```

Save back in data.

```
Z[[grep("h.*ass", names(Z))]] ← xha
saveRDS(xha, paste0(path1234, "HHAssetsCleaned.rds"))
saveRDS(completeAsset, paste0(path1234, "ListOfCompleteAssetsInAllRounds.rds"))
```

III.3.2 Productive assets

Productive asset contents.

```
# substitute contents to code
pcodecon ← read.table(text="rd contents code
1 'tractor' 401
1 'thresher' 402
1 'power tiller' 403
1 'power pump' 404
1 'deep and shallow tube-well' 405
1 'treddle pump' 406
1 'done/swing basket' 407
1 'plough and yoke' 408
1 'spray' 409
1 'husking machine' 410
1 'ginning machine' 411
1 'country boat' 412
1 'engine boat' 413
1 'fishing net' 414
1 'cage incubator' 415
1 'brooder' 416
1 'bees-box' 417
1 'weeder' 418
1 'ladder (moi)' 419
1 'sickle/dao/axe/spade' 420
1 'gola (grain storage)' 421
1 'saw' 422
1 'dheki' 423
1 'jata' 424
1 'rickshaw' 425
1 'other, specify' 426
2 'tractor' 401
2 'thresher' 402
2 'power tiller' 403
2 'power pump' 404
2 'hand pump' 405
2 'deep tube-well' 406
2 'shallow tube-well' 407
2 'treddle pump' 408
2 'rower pump' 409
2 'done/swing basket' 410
2 'plough and yoke' 411
2 'spray' 412
2 'husking machine' 413
2 'ginning machine' 414
2 'country boat' 415
2 'engine boat' 416
2 'fishing net' 417
2 'cage incubator' 418
2 'brooder' 419
2 'bees-box' 420
```

```
2 'weeder' 421
2 'ladder (moi)' 422
2 'sickle/dao/axe/spade' 423
2 'gola (grain storage)' 424
2 'dheki' 425
2 'jata' 426
2 'sewing machine' 427
2 'other, specify' 428"
, header = T)
pcodecon ← data.table(pcodecon)
pcodecon1 ← pcodecon[rd == 1, ]
pcodecon2 ← pcodecon[rd == 2, ]
xpa[, pa2code := pa2]
xpa[, pa3code := pa3]
xpa[, pa2 := as.character(pa2)]
xpa[, pa3 := as.character(pa3)]
xpa[, pa4 := tolower(as.character(pa4))]
for (ll in pcodecon1[, code])
  xpa[grepl(ll, pa2code) & survey == 1, pa2 := pcodecon1[code==ll, contents]]
for (ll in pcodecon2[, code])
  xpa[grepl(ll, pa2code) & survey ≥ 2, pa2 := pcodecon2[code==ll, contents]]
for (ll in pcodecon1[, code])
  xpa[grepl(ll, pa3code) & survey == 1, pa3 := pcodecon1[code==ll, contents]]
for (ll in pcodecon2[, code])
  xpa[grepl(ll, pa3code) & survey ≥ 2, pa3 := pcodecon2[code==ll, contents]]
xpa[, pa1 := tolower(pa1)]
# There are HHs in survey == 1 who report code 427 (sewing machine), 428 (other)
# which do not exist in the questionniare. Possibly an error in data entry. Use pcodecon2
for (ll in c(427, 428))
  xpa[grepl(ll, pa2code) & survey == 1, pa2 := pcodecon2[code==ll, contents]]
for (ll in c(427, 428))
  xpa[grepl(ll, pa3code) & survey == 1, pa3 := pcodecon2[code==ll, contents]]
# pa4 has typos
xpa[grepl("ladde", pa4), pa4 := "ladder(moi)"]
xpa[grepl("ladde", pa2), pa2 := "ladder(moi)"]
xpa[grepl("swing", pa4), pa4 := "sewing machine"]
xpa[grepl("other", pa4), pa4 := "other, specify"]
```

Productive asset records xpa is not an exhaustive list, and HHs with no productive asset are omitted in the file. Mean assets for household and productive assets.

	survey	meanPA	meanNarrowPA	medianPA	medianNarrowPA	stdPA
1:	1	1244.203	927.511	600	400	4651.05
2:	2	1267.192	965.669	400	380	6496.36
3:	3	1269.333	1050.763	430	420	5700.13
4:	4	840.997	773.076	400	400	4131.01

Save back in data.

```
Z[[grep("pr.*ass", names(Z))]] ← xpa
saveRDS(xpa, paste0(path1234, "ProdAssetsCleaned.rds"))
```

III.3.3 Land holding in contract and ownership file

```
lnd ← Z[[grep("contr", names(Z))]]
# Change NA to zero
for (i in 1:3)
```

```
  lnd[eval(parse(text = paste0("is.na(area_", i, ")"))), paste0("area_", i) := 0]
# OwnedArea = area_1[own_con_1=="Own"]
#  +area_2[own_con_2=="Own"]+area_3[own_con_3=="Own"]
lnd[, OwnedArea := 0]
for (i in 1:3)
  lnd[grepl("^Own$", eval(parse(text = paste0("own_con_", i)))),
    OwnedArea := OwnedArea + eval(parse(text = paste0("area_", i)))]
# OperatedArea = OwnedArea
#  + area_1[own_con_1=="rent|share|awne"]
#  + area_2[own_con_2=="rent|share|awne"]
#  + area_3[own_con_3=="rent|share|awne"]
lnd[, OperatedArea := OwnedArea]
for (i in 1:3)
  lnd[grepl("rent|share|awne", eval(parse(text = paste0("own_con_", i)))),
    OperatedArea := OperatedArea + eval(parse(text = paste0("area_", i)))]
destat(lnd[OwnedArea > 0 | OperatedArea > 0, .(year, survey, OwnedArea, OperatedArea)])
```

	min	25\\%	median	75\\%	max	mean	std	0s	NAs	n
year	2012	2014	2015	2015	2017	2014.9	1.2	0	0	753
survey	1	2	3	3	4	2.7	0.8	0	0	753
OwnedArea	0	0	0	2	748	9.9	42.0	425	0	753
OperatedArea	1	1	3	32	1155	24.0	78.4	0	0	753

```
round(lnd[, .(
  meanOwA = mean(OwnedArea),
  medOwA = median(as.numeric(OwnedArea)),
  NZeroOwA = sum(OwnedArea>0),
  stdOwA = var(OwnedArea)^(.5),
  meanOpA = mean(OperatedArea),
  medOpA = median(as.numeric(OperatedArea)),
  NZeroOpA = sum(OperatedArea>0),
  stdOpA = var(OperatedArea)^(.5)),
  by = survey][order(survey)], 3)
```

	survey	meanOwA	medOwA	NZeroOwA	stdOwA	meanOpA	medOpA	NZeroOpA	stdOpA
1:	1	0.240	0	11	3.858	0.659	0	35	6.063
2:	2	1.799	0	120	23.551	4.214	0	266	43.730
3:	3	1.151	0	131	8.765	2.694	0	307	19.270
4:	4	0.336	0	66	3.156	0.978	0	145	6.228

Land holding in ownership_and_contract file does not have most of round 1 information. Coverage is also limited to agricultural land.

	year				
OperatedLand	2012	2014	2015	2017	Sum
FALSE	778	680	668	662	2788
TRUE	18	79	91	46	234
Sum	796	759	759	708	3022

	year				
OwnedLand	2012	2014	2015	2017	Sum
FALSE	790	724	715	686	2915
TRUE	6	35	44	22	107
Sum	796	759	759	708	3022

Save back in data.

```
Z[[grep("contr", names(Z))]] ← lnd
```

III.3.4 Agricultural and residential land in HH Asset file

May 22, 2020 Land holding information has many missing values. Almost 1/2 of respondents do not reply. One should not rely too much on the estimated results because we do not know much about the sample selection. It is possible that NAs indicate zero's, because most of the char residents have little land holding.

Abu-san's email on Jan 30, 2020 I checked the questionnaire and found that from round 2, landholding information has been included in the asset information, which made the asset data inflated from round 2. Since landholding is something that is time-invariant for the ultra-poor households, either we can add the landholding information in round 1 or create an asset holding information deleting the landholding information from round 2 onwards, to make the valid comparison.

Land holding values do not vary much across survey rounds. Use purchase_in_last_1_year to reconstruct rd 1 residential land holding. For the households who report to have purchased land in last 1 year, assume baseline land holding to be zero.

There are 3 types of land reported. Use only aggregated total value. To do so, one needs to keep only the first type of land holding. In the file, all 3 types are stored so one needs to drop redundant entries whenever necessary.

```
# If purchased in last 1 year, assume that all value is acquired in last 1 year
# and set baseline amount to zero
hasL[Added & year ≤ 2013, amount := 0, by = hhid]
# Get total of: agricultural, residential, fallow/submerged land
hasL[, AmountFilled := sum(amount, na.rm = T), by = .(hhid, year)]
hasL[, EarliestAmount := AmountFilled[!is.na(AmountFilled)][1], by = hhid]
# EachAmountFilled is land values of various types
setnames(hasL, "amount", "EachAmountFilled")
# Multiple land holding entries because multiple land types. For total, keep only one type
saveRDS(hasL, paste0(pathsaveHere, "LandNAFilled.rds"))
```

Land holding of original 776 HHs. Only 458 households responded. We assume all other households to have zero land holding. This is possible if their residential land is rented.

```
        year
HaveLand 2012 2013 2014 2015 2017  Sum
   FALSE   35   35   21    5    1   97
   TRUE   423  423  437  453  457 2193
   Sum    458  458  458  458  458 2290
```

III.3.5 Merge HH and productive assets and land holding information in lnd

I merge household assets (NLHAssetAmount, BroadNLHAssetAmount, NLHAssetAmountOneYear, NLHAssetNum, BroadNLHAssetNum) with productive assets (PAssetAmount, PAssetEarning) together (merged data is called xas).

```
xha ← readRDS(paste0(path1234, "HHAssetsCleaned.rds"))
xpa ← readRDS(paste0(path1234, "ProdAssetsCleaned.rds"))
xpa[is.na(PAssetAmount), PAssetAmount := 0]
xpa[is.na(PAssetEarning), PAssetEarning := 0]
dit2 ← unique(xha[, .(hhid, year, survey,
  #NarrowNLHAssetAmount,
  NLHAssetAmount, BroadNLHAssetAmount,
  #RNarrowNLHAssetAmount, RNLHAssetAmount, RBroadNLHAssetAmount,
  NLHAssetAmountOneYear,
  #NarrowNLHAssetNum,
  NLHAssetNum, BroadNLHAssetNum
```

```
  )])
xas ← merge(dit2, xpa, by = c("hhid", "year", "survey"), all = T)
```

Tabulate number of households who report newly acquired household assets.

```
      bought.last1year
year    FALSE TRUE <NA>
  2012      0    0 1486
  2013      0    0  618
  2014   1698  168  205
  2015   1178  862   49
  2017   1240  755    9
```

```
   survey meanNLHA medianNLHA stdNLHA meanNumNLHA medianNumNLHA
1:      1  1397.75        1200  857.66     1.00000             1
2:      2  2077.45        1800 1667.68     1.39871             1
3:      3  2292.26        1800 2921.89     1.69657             2
4:      4  2249.91        1700 2774.64     1.78195             2
```

Productive asset items (total of all rounds, excluding livestock hence NL assets).

```
            bees-box              brooder       cage incubator
                6276                   75                  903
       country boat       deep tube well               dheki
                  17                   17                 1221
  done/swing basket          engine boat          fishing net
                   8                   11                  453
    ginning machine gola (grain storage)            hand pump
                1550                   25                  786
    husking machine                 jata          ladder(moi)
                  62                   51                   32
     other, specify      plough and yoke           power pump
                   8                    4                   13
       power tiller             rickshaw           rower pump
                  12                   25                    7
                 saw       sewing machine    shallow tube well
                  24                   10                    1
sickle/dao/axe/spade               spray             thresher
                 918                   26                    2
             tractor         treddle pump               weeder
                   5                   66                   30
               <NA>
              20434
```

Merge land holding data with asset data.

```
hasL ← readRDS(paste0(pathsaveHere, "LandNAFilled.rds"))
# keep only one row per hhid
hasL[, max := 1:.N, by = .(hhid, year)]
hasL ← hasL[max == 1, .(hhid, year, AmountFilled, Added)]
hasL[, max := NULL]
```

```
Warning in `[.data.table`(hasL, , `:=`(max, NULL)): Column 'max' does not exist to remove
```

```
#lnd ← Z[[grep("contr", names(Z))]]
#lnd0 ← lnd[OwnedArea > 0 | OperatedArea > 0, ]
commoncols ← intersect(colnames(hasL), colnames(xas))
xas2 ← merge(xas, hasL, by = commoncols, all.x = T)
#xas3 ← merge(xas, hasL, by = c("hhid", "year"), all.x = T)
```

Save data.

```
Z$MergedAssets ← xas2
fnd ← c(fnd, "MergedAssets")
saveRDS(xas3, paste0(path1234, "MergedAssetsCleaned.rds"))
```

III.3.6 Livestock

There are 3 entries with type == NA. For hhid 7126814, 7127105, they are cows. For 7043316, it sold calf, so fill cow/ox.

```
      hhid survey year type ownership own_share number_owned mrkt_value
1: 7043316      2 2014             Yes       own            1      10000
2: 7126814      3 2015             Yes       own            1      17000
3: 7127105      4 2017             Yes       own            1      28000
   sold_value sale_amount sold dead born base nowned_cow nowned_ox nowned_goat
1:      15000          NA    1   NA   NA   NA         NA        NA          NA
2:      12000          NA    1   NA   NA   NA         NA        NA          NA
3:      51000          NA    1   NA   NA   NA         NA        NA          NA
   u_id mid s8b_1 nowned_chicken nowned_sheep sales_cow sales_ox sales_sheep
1:   NA  NA    NA             NA           NA        NA       NA          NA
2:   NA  NA    NA             NA           NA        NA       NA          NA
3:   NA  NA    NA             NA           NA        NA       NA          NA
   _merge eaten labor_hired labor_hired_day labor_payment total_cost dup
1:   <NA>    NA          NA              NA            NA       2000   0
2:   <NA>    NA          NA              NA            NA       2000   0
3:   <NA>    NA          NA              NA            NA       2000   0
   counttime
1:         4
2:         4
3:         4
```

Number owned by hhid 7096201 in survey == 3 is 24000 while its market value is 1. Switch these entries.

In round 1, nowned_X is used for X = cow, ox, goat, sheep, chicken, and type, number_owned are used. For type==Goat/Sheep, only nowned_goat is copied to number_owned and nowned_sheep is not included.

```
           hhid survey       type nowned_cow nowned_goat nowned_sheep
  1:    7010102      1 Goat/Sheep          0           2            4
  2:    7010102      2 Goat/Sheep         NA          NA           NA
  3:    7010107      1 Goat/Sheep          0           1            3
  4:    7010108      1 Goat/Sheep          0           1            3
  5:    7010114      1 Goat/Sheep          0           1            3
 ---
201: 99070712709      4 Goat/Sheep         NA          NA           NA
202: 99070712714      4 Goat/Sheep         NA          NA           NA
203: 99081412504      1 Goat/Sheep          0           1            5
204: 99081711215      4 Goat/Sheep         NA          NA           NA
205: 99081912103      1 Goat/Sheep          0           1            1
     number_owned
  1:            2
  2:            1
  3:            1
  4:            1
  5:            1
 ---
201:            4
202:            3
203:            1
204:            4
205:            1
```

I will add sheep to number_owned. For cows and oxen, there is no mismatch in numbers reported between nowned_cow+nowned_ox and number_owend with type==cow/ox.

From round 2 onwards, only type, number_owned are used. Use the latter way to show information. To do so, reshape to long and reshape back. Other information mrkt_value, sold_amount, labor_X, total_cost are almost all NAs, so drop them.

```
      number_owned
survey   0   1   2   3   4   5   6   8   9 Sum
     1   0  22  40   4  16   6  10   1   1 100
     2  39   0   0   0   0   0   0   0   0  39
     3  26   0   0   0   0   0   0   0   0  26
     4  40   0   0   0   0   0   0   0   0  40
```

```
            hhid survey        type nowned_cow nowned_goat nowned_sheep
  1:     7010102      1 Goat/Sheep          0           2             4
  2:     7010102      2 Goat/Sheep         NA          NA            NA
  3:     7010107      1 Goat/Sheep          0           1             3
  4:     7010108      1 Goat/Sheep          0           1             3
  5:     7010114      1 Goat/Sheep          0           1             3
 ---
201: 99070712709      4 Goat/Sheep         NA          NA            NA
202: 99070712714      4 Goat/Sheep         NA          NA            NA
203: 99081412504      1 Goat/Sheep          0           1             5
204: 99081711215      4 Goat/Sheep         NA          NA            NA
205: 99081912103      1 Goat/Sheep          0           1             1
     number_owned
  1:            6
  2:            0
  3:            4
  4:            4
  5:            4
 ---
201:            0
202:            0
203:            6
204:            0
205:            2
```

```
      type
survey      Chicken/duck cow/ox Goat/Sheep  Sum
     1  727          837    553         100 2217
     2  257          413   1370          39 2079
     3   64          224   1803          26 2117
     4   82          229   1653          40 2004
```

```
            hhid survey   type number_owned sold dead born base nowned_cow
  1:     7010119      1 cow/ox            1    0    0    0    1          1
  2:     7010120      1 cow/ox            1    0    0    0    1          1
  3:     7020301      1 cow/ox            1    0    2    0    1          1
  4:     7020302      1 cow/ox            1    0    3    0    1          1
  5:     7020306      1 cow/ox            1    0    3    0    1          1
 ---
549: 99081912414      1 cow/ox            1    0    0    0    2          1
550: 99081912415      1 cow/ox            1    0    0    0    2          0
551: 99081912416      1 cow/ox            1    0    0    0    2          1
552: 99081912419      1 cow/ox            1    0    0    0    2          0
553: 99081912420      1 cow/ox            1    0    5    0    2          1
     nowned_ox nowned_goat nowned_sheep nowned_chicken
  1:         0           0           0              0
  2:         0           0           0              0
```

```
  3:          0            0            0              4
  4:          0            3            3              4
  5:          0            3            3             10
 ---
549:          0            0            0              3
550:          1            0            0              5
551:          0            0            0              4
552:          1            0            0              5
553:          0            0            0              7
```

```
Warning in `[.data.table`(xlo1L, , `:=`(LVcode, type)): Invalid .internal.selfref detected
```

```
      LVcode
survey chickenduck cowox goatsheep  Sum
   1          2217  2217       2217 6651
   Sum        2217  2217       2217 6651
```

Livestock holding of 800 HHs at round 1:

```
             number_owned
LVcode            0    1    2    3    4    5    6    7    8    9   10   11   12
  chickenduck   405   34  111   71   57   57   17    5   10    5   11    2    5
  cowox         654  102   31    6    2    1    0    0    0    0    0    0    0
  goatsheep     656   39   44   14   33    3    6    1    0    0    0    0    0
  Sum          1715  175  186   91   92   61   23    6   10    5   11    2    5
             number_owned
LVcode           13   15   22  Sum
  chickenduck     2    3    1  796
  cowox           0    0    0  796
  goatsheep       0    0    0  796
  Sum             2    3    1 2388
```

Livestock reported to be owned with zero self-evaluated market value. Need to impute values for these livestock. Median sales price of a cow/ox is 20000, a goat is 2533.333333. Use these to impute values ImputedValue. Cow prices vary a lot by years, so use annual prices for cows and call the imputed values as Imputed2Value.

```
xloL[, ImputedValue := cowprice * number_owned]
#xlo[grepl("ox", LVcode), ImputedValue := oxprice * number_owned]
xloL[grepl("oa", LVcode), ImputedValue := goatprice * number_owned]
xloL[grepl("duck", LVcode), ImputedValue := chickduckprice * number_owned]
xloL[!grepl("co|ox|oa", LVcode), ImputedValue := mrkt_value]
xloL[!grepl("co|ox|oa", LVcode), ImputedValue := mrkt_value]

for (yr in cowpriceByYear[, year])
  xloL[year == yr, Imputed2Value := cowpriceByYear[year == yr, medprice] * number_owned]
for (yr in unique(xloL[, year])[!unique(xloL[, year]) %in% cowpriceByYear[, year]])
  xloL[year == yr, Imputed2Value := cowprice * number_owned]
xloL[grepl("oa", LVcode), Imputed2Value := goatprice * number_owned]
xloL[!grepl("co|ox|oa", LVcode), Imputed2Value := mrkt_value]
```

TotalImputedValues:

```
# Livestock assets.
setkey(xloL, hhid, year, survey)
xloL[, TotalImputedValue := sum(ImputedValue, na.rm = T), by = list(hhid, year)]
xloL[, TotalImputed2Value := sum(Imputed2Value, na.rm = T), by = list(hhid, year)]
xloL[, TotalSelfEvaluatedValue := sum(mrkt_value, na.rm = T), by = list(hhid, year)]
destat(xloL[, .(TotalImputedValue, TotalImputed2Value)])
```

	min	25\\%	median	75\\%	max	mean	std	0s	NAs	n
TotalImputedValue	0	200	20000	40000	300000	21805.3	22017.9	5028	0	24042
TotalImputed2Value	0	200	25000	40000	395000	30491.0	33842.4	5028	0	24042

Livestock values at baseline among 800 HHs:

	number_owned									
PositiveLivestockAtBaseline	0	1	2	3	4	5	6	7	8	9
FALSE	539	17	61	46	41	30	11	2	9	3
TRUE	242	141	111	42	49	31	12	4	1	2
Sum	781	158	172	88	90	61	23	6	10	5

	number_owned						
PositiveLivestockAtBaseline	10	11	12	13	15	22	Sum
FALSE	4	2	2	1	1	1	770
TRUE	7	0	3	1	2	0	648
Sum	11	2	5	2	3	1	1418

Cows owned at the baseline (dummyHadCows).

	dummyHadCows		
survey	0	1	Sum
1	6099	552	6651
2	5466	0	5466
3	6159	0	6159
4	5766	0	5766

	dummyHadCows			
NoBaseline	0	1	<NA>	Sum
0	17565	6144	0	23709
1	0	0	333	333

	dummyHadCows			
survey	0	1	<NA>	Sum
1	4995	1656	0	6651
2	3903	1482	81	5466
3	4455	1542	162	6159
4	4212	1464	90	5766

	LVcode			
survey	chickenduck	cowox	goatsheep	Sum
1	2217	2217	2217	6651
2	1822	1822	1822	5466
3	2053	2053	2053	6159
4	1922	1922	1922	5766

The number of cows owned:

	NumCows												
survey	0	1	2	3	4	5	6	7	8	9	10	12	15	<NA>
1	1664	385	139	21	4	2	1	0	0	0	0	0	0	1
2	11	688	396	125	36	15	4	1	2	2	0	0	1	541
3	11	1011	506	134	50	10	6	4	4	0	0	1	0	316
4	17	745	664	157	37	12	7	1	2	1	3	0	0	276

	NumCows
survey	Sum
1	2217
2	1822
3	2053
4	1922

Number of observations per round and livestock.

```
      LVcode
survey chickenduck cowox goatsheep  Sum
     1        2217  2217      2217 6651
     2        1822  1822      1822 5466
     3        2053  2053      2053 6159
     4        1922  1922      1922 5766
```

Reshape back livestock data to wide. Format like: ID, number.Cow, number.Goat, etc. (code not shown)

```
 [1] "hhid"                      "u_id"
 [3] "mid"                       "survey"
 [5] "year"                      "Imputed2Value"
 [7] "TotalImputedValue"         "TotalImputed2Value"
 [9] "TotalSelfEvaluatedValue"   "dummyHadCows"
[11] "NoBaseline"                "NumCowsOwnedAtRd1"
[13] "NumCows"                   "NumOwned.cowox"
[15] "Ownership.cowox"           "OwnShare.cowox"
[17] "MktValue.cowox"            "SoldValue.cowox"
[19] "Sold.cowox"                "Dead.cowox"
[21] "Born.cowox"                "Eaten.cowox"
[23] "TotalCosts.cowox"          "ImputedValue.cowox"
[25] "NumOwned.chickenduck"      "Ownership.chickenduck"
[27] "OwnShare.chickenduck"      "MktValue.chickenduck"
[29] "SoldValue.chickenduck"     "Sold.chickenduck"
[31] "Dead.chickenduck"          "Born.chickenduck"
[33] "Eaten.chickenduck"         "TotalCosts.chickenduck"
[35] "ImputedValue.chickenduck" "NumOwned.goatsheep"
[37] "Ownership.goatsheep"       "OwnShare.goatsheep"
[39] "MktValue.goatsheep"        "SoldValue.goatsheep"
[41] "Sold.goatsheep"            "Dead.goatsheep"
[43] "Born.goatsheep"            "Eaten.goatsheep"
[45] "TotalCosts.goatsheep"      "ImputedValue.goatsheep"
```

In 2012, 2013, price is not unit price but actually the total sales. Sales prices of livestock produce are recorded as in Figure 7, by correcting errors in egg and milk prices in 2014, we get Figure 8.

Create a variable ImputedPrice: total_sold_tk for 2012, 2013, total_sold_tk/total_sold for 2014, 2015, 2017.

```
        UnitPrice
produce     8     9    10    30    40    50 <NA>  Sum
   egg     20     1   270     0     0     0  329  620
   milk     0     0     0     7    32    98  483  620
   Sum     20     1   270     7    32    98  812 1240
```

Correct produce prices: For milk prices above 200, we use median price of below 200. Same for egg prices with a threshold of 15.

Save in the original list.

```
Z[[grep("liv.*ow", names(Z))]] ← xloW
Z[[grep("liv.*pr", names(Z))]] ← xlp2
Z$LivestockLong ← xloL
fnd ← c(fnd, "LivestockLong")
```

III.4 Poverty

III.4.1 Monga

Correct some typos in monga_meals (not shown). Visualise monga period meals per day (Figure 21). Save in the original list.

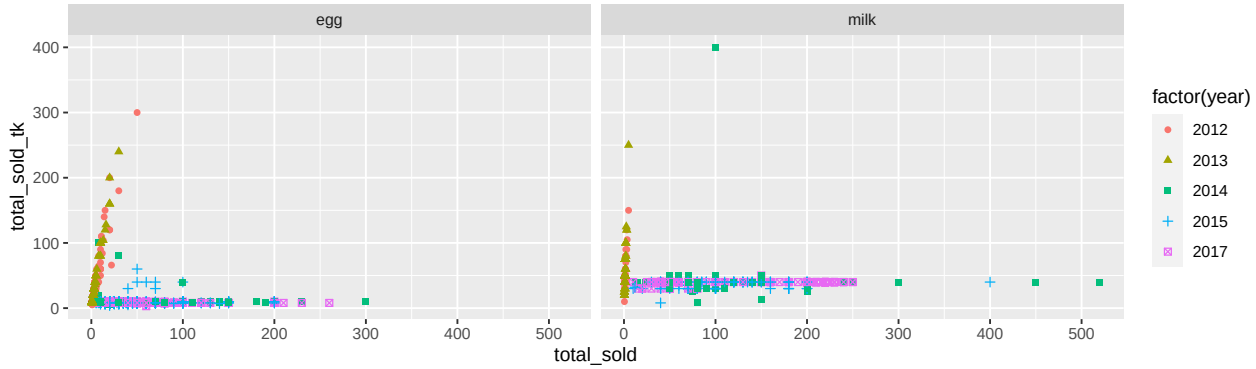

Figure 7 Produce sales raw prices

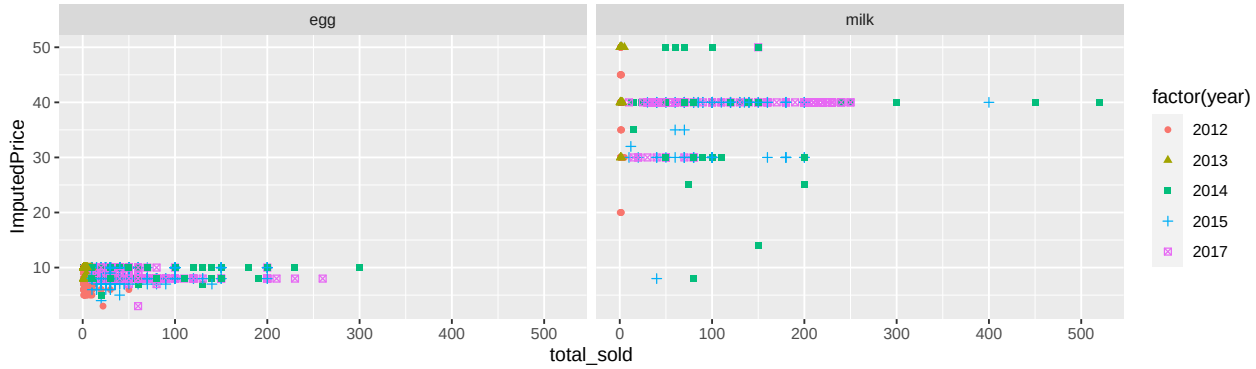

Figure 8 Produce sales corrected prices

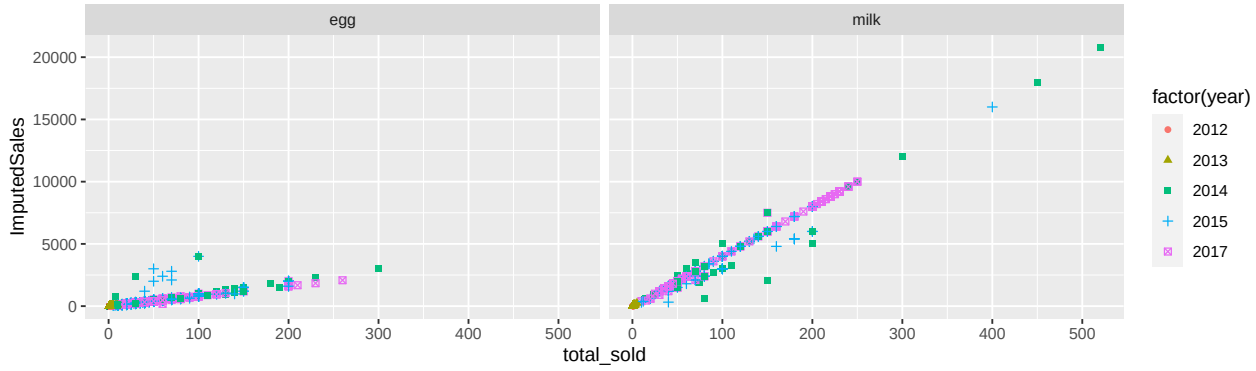

Figure 9 Imputed produce sales using corrected prices

```
Z[[grep("mong", names(Z))]] ← xm
```

III.4.2 Saving

Saving is given in Figure 22.

III.5 Other

Superfluous entries in farm_production, ownership_and_contract, poverty_updated, poverty. There are 7709, 0, 107, 133 rows with all NAs, respectively. Drop them (or otherwise this will cause many HHs without records in ID files while entries in section files). (These will cause nomber of

rows to be small as indicated in Table 2.)

```
Z[[grep("pov", names(Z))[1]]] ← Z[[grep("pov", names(Z))[1]]][!is.na(year), ]
Z[[grep("pov", names(Z))[2]]] ← Z[[grep("pov", names(Z))[2]]][!is.na(year), ]
fa ← Z[[grepout("far", names(Z))]]
ii ← apply(is.na(fa[,
        grep("area_1", colnames(fa)):grep("pri.*nit\\_3", colnames(fa)), with = F]), 1, a
Z[[grepout("far", names(Z))]] ← fa[!ii, ]
cont ← Z[[grepout("contr", names(Z))]]
cont ← a2b.data.table(cont, "", NA)
cont ← a2b.data.table(cont, "\\.", NA)
ii ← apply(is.na(cont[,
        grep("area_1", colnames(cont)):grep("sh.*ge\\_3", colnames(cont)), with = F]), 1,
Z[[grepout("contr", names(Z))]] ← cont[!ii, ]
```

Correct errors in survey numbering.

```
Z[[grep("credit", names(Z))]] ← Z[[grep("credit", names(Z))]][year == 2017, survey := 4]
Z[[grep("labo", names(Z))]] ← Z[[grep("labo", names(Z))]][year == 2013, survey := 1]
```

Convert empty space or dot to NA.

```
cr ← Z[[grepout("borr", names(Z))]]
cr ← a2b.data.table(cr, "", NA)
Z[[grepout("borr", names(Z))]] ← cr
xla ← Z[[grepout("lab", names(Z))]]
setkey(xla, hhid, year)
xla ← a2b.data.table(xla, "", NA)
Z[[grepout("lab", names(Z))]] ← xla
```

Save all data.

```
saveRDS(Z, paste0(path1234, "data_read_in_a_list_2.rds"))
```

IV Attach treatment information to all files

Attach xid to all files other than roster.

```
Z ← readRDS(paste0(path1234, "data_read_in_a_list_2.rds"))
xid ← readRDS(paste0(path1234, "ID.rds"))

xid2 ← unique(xid[, .(gid, hhid, povertystatus, year, survey,
        ObPattern, AttritIn,
        membership, Mstatus, Mpattern, Mgroup, Assign, Hhidyear,
   randomization, AssignOriginal, AssignRegression,
   IntDate, DistDate1, DistDate2, DistDate3,
   Disbursed1, Disbursed2, Disbursed3)])
```

Attach treatment info in admin-roster to each files. For education file, also attach roster.

```
for (i in c(2, 1, 3:length(Z))) {
# this flip order in 1, 2 makes merge operation of edu with roster easier
        if (any(which(grepl("id\\_", names(Z))) %in% i)) next
        x1 ← Z[[i]]
        #\textsf{lonely.hhid}: Found in ID but not in `section' page (here, labour income)
        table(lonely.hhid ← unique(x1[, hhid]) %in% xid[, hhid])
        lonely.hhid ← unique(x1[, hhid])[!lonely.hhid]
        x2 = copy(x1[!(hhid %in% lonely.hhid), ])
        if (any(grepl("yearRoster", colnames(x2)))) setnames(x2, "yearRoster", "year")
```

```
        if (any(grepl("survey", colnames(x2))))
                setkey(x2, hhid, year, survey) else
                setkey(x2, hhid, year)
        x3 <- xid2[x2]
        Z[[i]] <- x3
        # for education, attach entire roster file
        if (any(which(grepl("edu", names(Z))) %in% i)) {
                x1 <- Z[[i]]
                ros <- Z[[grep("ros", names(Z))]]
                lonely.hhid <- unique(x1[, hhid]) %in% ros[, hhid]
                lonely.hhid <- unique(x1[, hhid])[!lonely.hhid]
                x2 = copy(x1[!(hhid %in% lonely.hhid), ])
                if (any(grepl("hhidy", colnames(ros))))
                        ros[, grepout("hhidy", colnames(ros)) := NULL]
                # need to merge with key = survey (not year)
                # because ros has only 2012 or survey == 1
                setnames(ros, "year", "yearRoster")
                setkey(ros, hhid, mid, survey)
                setkey(x2, hhid, mid, survey)
                x3 <- ros[x2]
                firstcols <- c("gid", "hhid", "mid", "survey", "year", "yearRoster",
                        colnames(x2)[!grepl("hhid$|mid|gid|survey|^i?.?year$", colnames(x2
                setcolorder(x3, c(firstcols, colnames(ros)[!(colnames(ros)%in%firstcols)])
                setkey(x3, hhid, mid, year, survey)
                Z[[i]] <- x3
        }
}
```

Save all data.

```
saveRDS(Z, paste0(path1234, "data_read_in_a_list_with_treatment.rds"))
```

V Attrition

I define attrition as attriting in any rds 2, 3, 4 other than rejection or erosion.

```
Z <- readRDS(paste0(path1234, "data_read_in_a_list_with_treatment.rds"))
xid <- readRDS(paste0(path1234, "ID.rds"))
ros <- Z[[grep("rost", names(Z))]]  # roster
xas2 <- readRDS(paste0(path1234, "MergedAssetsCleaned.rds"))
xid1 <- unique(xid[, .(hhid, AssignOriginal)])
```

There are 78 cases of attrition out of 2022 non-dropout HHs in sample. Plot characteristics of attrited HHs in Figure 10. As can be seen, nothing seems to differ across arms for attrited HHs.

Probit regression using all sampled HHs reveals attrition is random, not systematic, for arm assignment (Table 1). Covariates are HeadAge, HHsize, HAssetAmount, PAssetAmount, OwnedArea, OperatedArea, AssignOriginal. Not surprisingly, larger Owned area reduces attrition as the wealthier individuals live in a better condition and is less likely to be affected by flood or economic shocks.

VI Further correction using merged information

```
Z <- readRDS(paste0(path1234, "data_read_in_a_list_with_treatment.rds"))
xid <- readRDS(paste0(path1234, "ID.rds"))
```

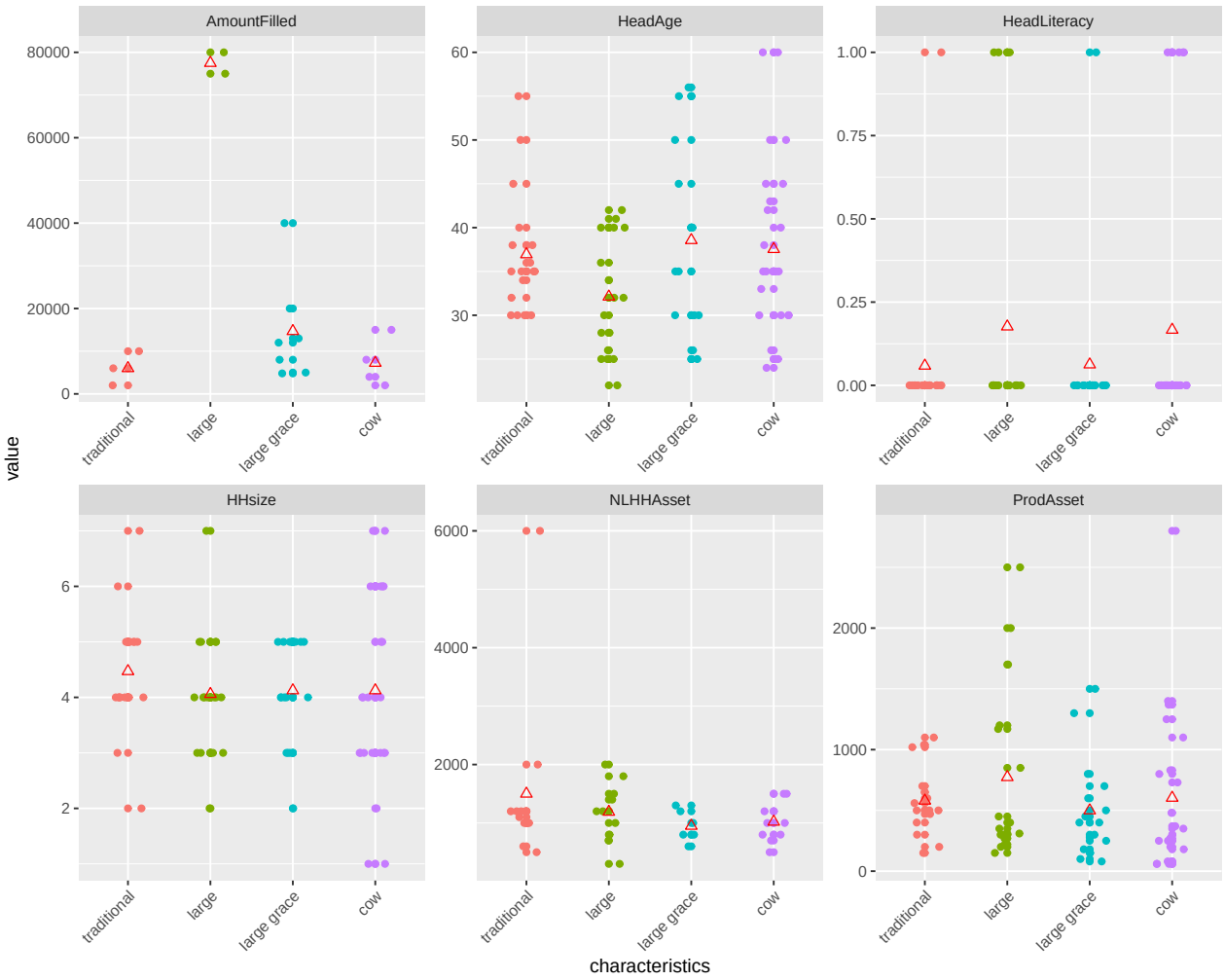

Figure 10 Characteristics of attrited HHs between arms

TABLE 1: ATTRITION SELECTION

variables	(1)	(2)
(Intercept)	−1.565***	−0.173
	(0.0)	(0.6)
HeadAge	−0.009	−0.016***
	(0.1)	(0.0)
Household size	0.046	0.076
	(0.2)	(0.1)
Household asset (taka)	−0.000	−0.000
	(0.2)	(0.7)
Productive asset (taka)	−0.000	−0.000***
	(0.2)	(0.0)
Owned area (decimal)	−0.392***	−0.133***
	(0.0)	(0.0)
Operating area (decimal)	0.000	−0.000
	(1.0)	(1.0)
large arm	0.119	
	(0.6)	
large grace arm	0.088	
	(0.7)	
cow arm	0.235	
	(0.3)	
group fixed effects		yes
mean of dependent variable	0.033	0.033
n	1947	2166

Source: Compiled from survey data, rounds 1 - 4.

Notes: 1. group fixed-effects are dummy variables for borrowing groups.

2. p values are shown in parentheses. $*$, $**$, $***$ indicate p values of 10%, 5%, 1%, respectively. Standard errors are clustered at the group level.

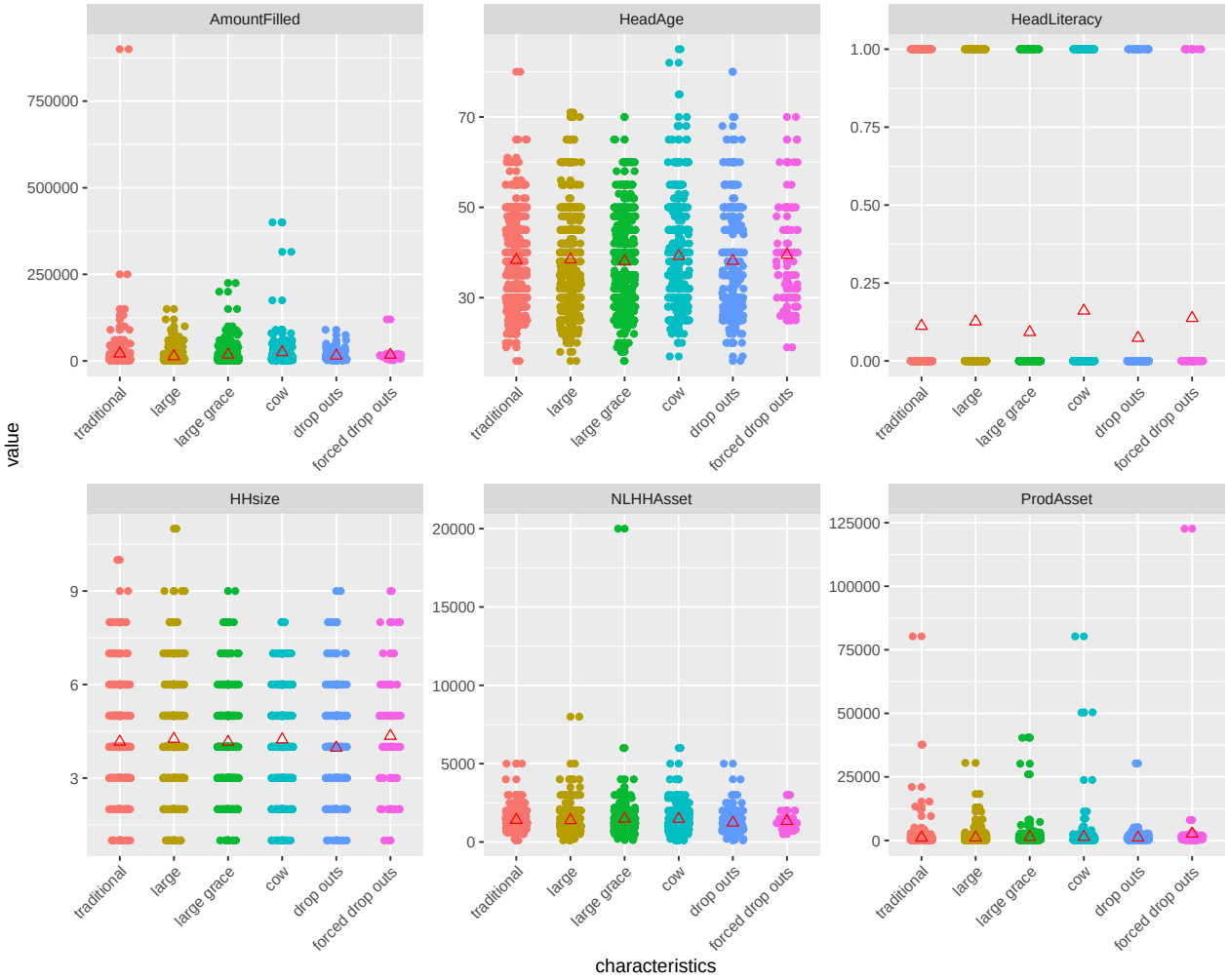

Figure 11 Characteristics of non-attrited HHs between arms

VI.1 Schooling

Below tabulation shows the number of times an individual is observed. Since we have 4 rounds, there are 6 duplicated entries which are dropped for now but are in need of correction.

```
   1    2    3    4    5
5795  669  728 2075    6
```

Here are the duplicated entries.

```
          hhid mid  memname En en year Age              primary
 1:    7031803   4  anrwara  1  1 2012  11                   94
 2:    7031803   4  anowara  2  2 2014  NA           unemployed
 3:    7031803   4  anowara  3  3 2015  NA           unemployed
 4:    7031803   4  anowara  4  3 2015  NA           unemployed
 5:    7031803   4  anowara  5  4 2017  NA           unemployed
 6:    7042220   4  rufikul  1  1 2012   4                   92
 7:    7042220   4  rofikul  2  2 2014  NA                child
 8:    7042220   4  rofikul  3  2 2014  NA                child
 9:    7042220   4  rofikul  4  3 2015  NA              student
10:    7042220   4  rofikul  5  4 2017  NA              student
11:    7054202   5    rubel  1  1 2012   4                   92
12:    7054202   5    rubel  2  2 2014  NA                child
13:    7054202   5    rubel  3  3 2015  NA              student
14:    7054202   5    rubel  4  3 2015  NA              student
```

```
15:     7054202   5     rubel   5   4 2017   NA                  student
16:     7065315   6    robina   1   1 2012    5                       92
17:     7065315   6    robina   2   2 2014   NA                  student
18:     7065315   6    robina   3   2 2014   NA                  student
19:     7065315   6    robena   4   3 2015   NA                  student
20:     7065315   6    robina   5   4 2017   NA                  student
21:     8169511   5 ayeguddi   1   1 2012   14                      94
22:     8169511   5  ayguddi   2   2 2014   NA              unemployed
23:     8169511   5  ayguddi   3   2 2014   NA              unemployed
24:     8169511   5  ayguddi   4   3 2015   NA agriculture wage labor
25:     8169511   5  ayguddi   5   4 2017   NA agriculture wage labor
26:  9807065208   5    parul   1   1 2013   10                      93
27:  9807065208   5    parul   2   2 2014   NA                  student
28:  9807065208   5    parul   3   3 2015   NA                  student
29:  9807065208   5    parul   4   3 2015   NA                  student
30:  9807065208   5    parul   5   4 2017   NA                  student
           hhid mid  memname En en year Age                  primary
```

Drop these.

```
xe ← xe[N < 5, ]
```

Check for further duplication when gid, hhid, mid, year, AgeComputed are used as an index.

```
[1] 7 duplicated entries.
```

```
    duplicated.num         hhid mid memname year AgeComputed currently_enrolled
 1:              1      7042417   5 ibrahim 2015           1                  No
 2:              1      7042417   5 ibrahim 2015           1                  No
 3:              4      7065006   4 somaiya 2017           7                 Yes
 4:              4      7065006   4 somaiya 2017           7                 Yes
 5:              6      7137302   3   kalam 2014          17                 Yes
 6:              6      7137302   3   kalam 2014          17                 Yes
 7:              7      8159115   4    sumi 2017           6                 Yes
 8:              7      8159115   4    sumi 2017           6                 Yes
 9:              5   9907065112   3   sajib 2014          13                 Yes
10:              5   9907065112   3   sajib 2014          13                 Yes
11:              2 99070511006   2  sohida 2013          42              <NA>
12:              2 99070511006   2  sohida 2013          42              <NA>
13:              3 99070511006   3     jui 2013           0                  No
14:              3 99070511006   3     jui 2013           0              <NA>
[1] Dropped 7 duplicated obs in edu file.
```

NAs in age. Use any age_1 info to fill NAs in age (process not shown).

NAs in sex: 9 individuals remain after copying whenever possible from other rounds. These individuals are dropped for now but are in need of correction for sex, age information.
[2017-11-14 Abu email]: A correction file is received. → (Yet to be applied.)

```
           hhid mid memname  sex age year
1:      7010112   5    <NA> <NA>  NA 2015
2:      7042614  NA    <NA> <NA>  NA 2017
3:      7043309  NA    <NA> <NA>  NA 2015
4:      7054310  NA    <NA> <NA>  NA 2015
5:      8169814   5    <NA> <NA>  NA 2012
6:   9807054106   3    <NA> <NA>  NA 2017
7:   9807054319   5    <NA> <NA>  NA 2013
8:   9807106517  NA    <NA> <NA>  NA 2015
9: 99070712714   5    <NA> <NA>  NA 2013
```

```
xe ← xe[!is.na(sex), ]
```

NAs in currently_enrolled. Below tabulation shows the primary occupation of these observations.

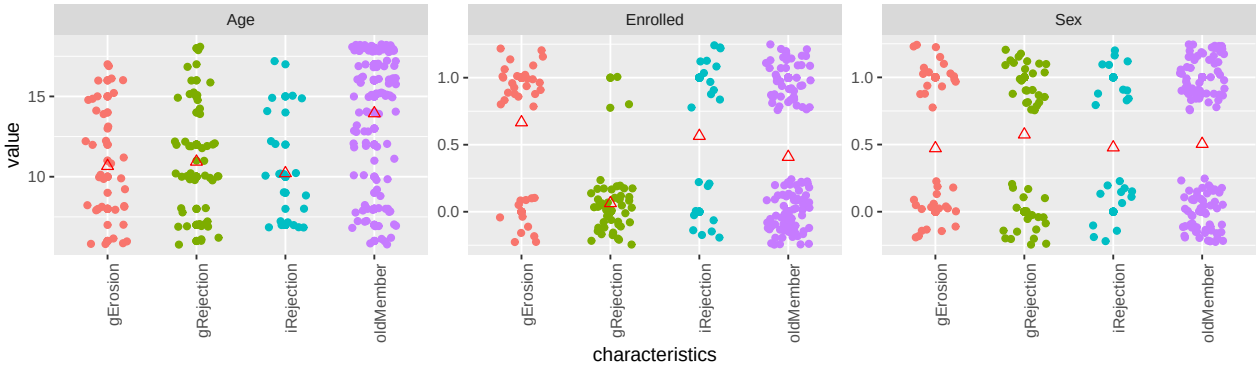

Figure 12 Characteristics of attrited members between arms

```
[1] 92 93 96 94 4  11 55
107 Levels:  advocate / moktar  house repairing (fixing) 1 11 13 16 17 18 ... wage labor i
```

Set currently_enrolled to "No" either if: primary == housewife, rel_hhh == spouse, edu == {never been to school, pre-school going age, pre-school, pre-madrasa}.

[2017-11-14 Abu email]: A correction file is received. → Corrected. Correction:

Having done so, below remains as NAs because these provide no clue on enrollment status. Need to be supplemented with new information on currently_enrolled. Drop these individuals for entire period for now.

```
       hhid mid year AgeComputed                        edu primary
1:  7021316   4 2012           16  class 5/finished class 5      96
2:  7042016   4 2012           18  class 5/finished class 5      96
3: 81710106   4 2012           18       finished ssc/dakhil      93
4: 81710516   6 2012           10 class 3/finished class 3       93
```

```
xe ← xe[!(Hhidmid %in%
        xe[AgeComputed ≥ 5 & AgeComputed ≤ 18 & is.na(currently_enrolled), Hhidmid]), ]
```

Define Enrolled: 1 if currently_enrolled is yes, 0 otherwise.

```
        year
Enrolled 2012 2013 2014 2015 2017
       0  710  172  355  216  114
       1 1388  634 1927 1808 1683
```

Why are there so many observations in 2012 and 2013 combined? Attrition in oldmember: 156 out of 355 reduction in obs is explained by iRejection, gErosion, gRejection.

```
      Mstatus
year   gErosion gRejection iRejection iReplacement newGroup oldMember <NA>
  2012      100         187         176            0        0      1630    5
  2013        0           0           0          184      622         0    0
  2014       56         121         130          164      536      1275    0
  2015       53         103         115          140      480      1133    0
  2017        0           0         208          120      438      1031    0
```

Plot characteristics of attrited members in Figure 12.

Define Schooling according to AgeComputed. This variable is time-variant. Enrollment by age at first observation (years 2012, 2013) is tabulated in below. We defined such that Enrolled==currently_enrolled for ages 5-18.

```
                  Age_1
currently_enrolled   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
```

```
                No      2    5    3    8    8 264 164   95   64   22   61   16   78   33   34
                Yes     0    0    1    2    4  12  52 262 305 153 390 121 238 131 116
                <NA> 167 166 210 234 263    0    0    0    0    0    0    0    0    0    0
                    Age_1
currently_enrolled  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29
                No  102  91  29  93   3   1   0   0   0   1   0   0   0   0   0
                Yes 129  58  28  39   1   1   0   0   0   0   3   0   0   1   0
                <NA>  0   0   0   0 108 227  41 156  72  80 388 106  78 276  47
```

```
         Age_1
Enrolled   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
       0 169 171 213 242 271 264 164  95  64  22  61  16  78  33  34 102  91
       1   0   0   1   2   4  12  52 262 305 153 390 121 238 131 116 129  58
         Age_1
Enrolled  17  18  19  20  21  22  23  24  25  26  27  28  29
       0  29  93 111 228  41 156  72  81 388 106  78 276  47
       1  28  39   1   1   0   0   0   0   3   0   0   1   0
```

Enrollment rates for ages 5 - 18 (Figure 23). Note that females can marry out which may be a reason for the general upward trend in female schooling.

Save schooling data.

```
saveRDS(xe, paste0(path1234, "schooling.rds"))
```

Use augmented panel to keep the denominator as the number of people in the first round, assuming that attrited females are not going to schools (need to check with roster updates). Without it, enrollment rates for females is inflated through time. This requires to generate NAs in missing rounds.

☞ Use data.table:::dcast to reshape to wide format which fills in NA for attrited members.

```
[1] 17585    88
```

```
       AgeComp
Age1    FALSE  TRUE
  FALSE 17296     0
  TRUE      0   289
```

```
[1] 9007   14
```

```
  1    2    3    4
515  337  390 1662
```

To a wide format and compute age using other years.

```
Xw2 ← dcast(Xw, ... ~ year, value.var =
        grepout("En|Age|^en$|^curr|edu", colnames(Xw)), sep = ".")
```

There are 806 individuals with AgeComputed==NA. All of these individuals have age information in other years.

```
             AgeComputed.2012
Enrolled.2013   6   7   8   9  10  11  12  13  14  15  16  17  18 <NA>
         0      0   0   0   0   0   0   0   0   0   0   0   0   0  172
         1      0   0   0   0   0   0   0   0   0   0   0   0   0  634
         <NA> 168 264 279 114 333  77 237 109 104 173 103  43  94    2
```

```
          AgeNAOtherYear
AgeNA2012 FALSE  TRUE
    FALSE  1665   433
    TRUE    808     0
```

There are 808 individuals whose age can be imputed from other rds. Impute (process not shown). Reshape back to long.

```
for (i in colnames(Xw2[, -c("hhid", "mid"), with = F]))
  Xw2[hhid == 9808148220, (i) := eval(parse(text =
    paste0(i, "[!is.na(", i, ")][1]")
    )), by = .(hhid, mid)]
Xw2 ← Xw2[!duplicated(Xw2), ]
# reshaping to long to fill in ages and GradeYear
X2 ← reshape(Xw2, direction = "long", idvar = c("hhid", "mid"),
        varying = grepout("\\.20", colnames(Xw2)))
setnames(X2, "time", "year")
setkey(X2, gid, hhid, mid, year)
X2[, tee := 1:.N, by = .(hhid, mid)]
```

```
Warning in `[.data.table`(X2, , `:=`(tee, 1:.N), by = .(hhid, mid)): Invalid .internal.sel
```

```
X2[AgeComputed ≥ 5 & AgeComputed ≤ 12, Schooling := "primary0512"]
X2[AgeComputed ≥ 13 & AgeComputed ≤ 15, Schooling := "junior1315"]
X2[AgeComputed ≥ 16 & AgeComputed ≤ 18, Schooling := "high1618"]
X2[, Schooling := factor(Schooling, levels =
        c("primary0512", "junior1315", "high1618"))]
```

Define GradeYear: Class grade in numerics.

```
       0   1   2   3   4   5   6   7   8  9 10 11 <NA> NonNATotal Total
2012 725 339 277 212 172 151  66  43  30 25 37 21  806       2098  2904
2013 102  97 109  80  79  73  48  29  27 15 21  0 2224        680  2904
2014   0 329 357 372 259 216 126 111  67 34 18  0 1015       1889  2904
2015   0 124 350 350 317 244 151 109  95 63 30  0 1071       1833  2904
2017   0  26  61 195 340 304 275 165 124 83 90  0 1241       1663  2904
```

Enrollment in comlete panel data:

```
      2012 2013 2014 2015 2017 total    1    2    3    4 total
0      710  172  355  216  114  1567  882  376  218   91  1567
1     1388  634 1927 1808 1683  7440 2022 2013 1834 1571  7440
total 2098  806 2282 2024 1797  9007 2904 2389 2052 1662  9007
```

Enrollment in augmented data: See unenrolled obs are added.

```
      2012 2013 2014 2015 2017 total    1    2    3    4    5 total
0     1516 2270  977 1096 1221  7080 1516 2270  977 1096 1221  7080
1     1388  634 1927 1808 1683  7440 1388  634 1927 1808 1683  7440
total 2904 2904 2904 2904 2904 14520 2904 2904 2904 2904 2904 14520
```

Define Year in augmented schooling panel: year 2013 → 2012.

```
        Year
Enrolled 2012 2014 2015 2017
       0 3786  977 1096 1221
       1 2022 1927 1808 1683
```

```
# forced dropouts
X2[Year == 2017 & grepl("for", AssignRegression), Enrolled := NA]
```

```
X2[, NumEnrollment := sum(Enrolled), by = list(AssignRegression, Schooling, sex, Year)]
X2[, NumberObs := .N, by = list(AssignRegression, Schooling, sex, Year)]
X2[, BaseNumber := NumberObs[1], by = list(AssignRegression, Schooling, sex)]
setkey(X2, hhid, mid, year)
X2[, tee := 1:.N, by = .(hhid, mid)]
```

```
      2012 2013 2014 2015 2017 total    1    2    3    4    5 total
0     1516 2270  977 1096 1121  6980 1516 2270  977 1096 1121  6980
1     1388  634 1927 1808 1683  7440 1388  634 1927 1808 1683  7440
<NA>     0    0    0    0  100   100    0    0    0    0  100   100
total 2904 2904 2904 2904 2904 14520 2904 2904 2904 2904 2904 14520
```

Save X1. Save attrition-augmented panel schooling data X2.

```
saveRDS(X1, paste0(path1234, "schooling_Age6-18InRd1.rds"))
saveRDS(X2, paste0(path1234, "schooling_augmented_panel.rds"))
```

VI.2 Missing ID file entries

```
     gid    hhid povertystatus year survey ObPattern AttritIn      membership
1: 70314 7031401          <NA> 2012      1      1000        2 Drop-out member
2:  <NA> 7031401          <NA> 2014      2      <NA>       NA            <NA>
      Mstatus Mpattern    Mgroup    Assign randomization AssignOriginal
1: gRejection     daaa drop outs drop outs          <NA>           <NA>
2:       <NA>     <NA>      <NA>      <NA>          <NA>           <NA>
   AssignRegression IntDate DistDate1 DistDate2 DistDate3 Disbursed1 Disbursed2
1:         dropOuts    <NA>      <NA>      <NA>      <NA>      FALSE         NA
2:             <NA>    <NA>      <NA>      <NA>      <NA>         NA         NA
   Disbursed3 base                          code_1 assist_1 cash_1 rice_kg_1
1:         NA    1                           <NA>       No     NA         NA
2:         NA   NA Remittance from family Members      Yes     NA         10
   tk_kg_1 wheat_flour_kg_1 wheat_flour_tk_kg_1 livestock_type_1 livestock_no_1
1:      NA               NA                  NA               NA            NA
2:      30               NA                  NA               NA            NA
   livestock_value_1 other_food_1 other_in_kind_1 compare_1
1:                NA           NA              NA      <NA>
2:                NA           NA              NA      <NA>
                                  code_2 assist_2 cash_2 rice_kg_2 tk_kg_2
1:                                  <NA>     <NA>     NA         NA       NA
2: Govt Scholarship for Primary Students      Yes   1200         NA       NA
   wheat_flour_kg_2 wheat_flour_tk_kg_2 livestock_type_2 livestock_no_2
1:               NA                  NA               NA            NA
2:               NA                  NA               NA            NA
   livestock_value_2 other_food_2 other_in_kind_2 compare_2 code_3 assist_3
1:                NA           NA              NA      <NA>   <NA>     <NA>
2:                NA           NA              NA      <NA>   <NA>     <NA>
   cash_3 rice_kg_3 tk_kg_3 wheat_flour_kg_3 wheat_flour_tk_kg_3
1:     NA        NA      NA               NA                  NA
2:     NA        NA      NA               NA                  NA
   livestock_type_3 livestock_no_3 livestock_value_3 other_food_3
1:               NA            NA                NA           NA
2:               NA            NA                NA           NA
   other_in_kind_3 compare_3 code_4 assist_4 cash_4 rice_kg_4 tk_kg_4
1:              NA      <NA>   <NA>     <NA>     NA         NA       NA
2:              NA      <NA>   <NA>     <NA>     NA         NA       NA
   wheat_flour_kg_4 wheat_flour_tk_kg_4 livestock_type_4 livestock_no_4
1:               NA                  NA               NA            NA
2:               NA                  NA               NA            NA
   livestock_value_4 other_food_4 other_in_kind_4 compare_4 dup pay_1 pay_2
```

```
1:                NA           NA               NA       <NA>    0     NA     NA
2:                NA           NA               NA       <NA>   NA     NA     NA
   pay_3 pay_4 counttime     Hhidyear
1:    NA    NA         2 7031401-2012
2:    NA    NA         2 7031401-2014
```

```
     gid    hhid povertystatus year survey    Mstatus Mpattern    Mgroup
1: 70314 7031401           <NA> 2012      1 gRejection     daaa drop outs
        membership    Assign     Hhidyear AssignRegression IntDate DistDate1
1: Drop-out member drop outs 7031401-2012         dropOuts    <NA>      <NA>
   DistDate2 DistDate3 Disbursed1 Disbursed2 Disbursed3
1:      <NA>      <NA>      FALSE          NA          NA
```

Some HHs have different years recorded in section files than ID file. For example, hhid==7137219 has 2012, 2014 in ID but 2015 in credit_and_borrowing, farm_production.

```
      hhid year    Mgroup    Assign      membership AssignOriginal Mpattern
1: 7137219 2012 drop outs drop outs Drop-out member    traditional     ddaa
2: 7137219 2014 drop outs drop outs Drop-out member    traditional     ddaa
```

```
      hhid year survey                filename
1: 7137219 2015      3        farm_production
2: 7137219 2015      3 ownership_and_contract
```

Below is the list of hhid and year that are missing in ID files. Why are these entries missing in ID file? Below gives the hhids.

```
  [1] 7020312-2014     7020412-2014     7021218-2014     7021220-2015
  [5] 7021320-2014     7031401-2014     7031401-2015     7031402-2014
  [9] 7031402-2015     7031403-2014     7031403-2015     7031404-2014
 [13] 7031404-2015     7031405-2014     7031405-2015     7031406-2014
 [17] 7031406-2015     7031407-2014     7031407-2015     7031408-2014
 [21] 7031408-2015     7031409-2014     7031409-2015     7031410-2014
 [25] 7031410-2015     7031411-2014     7031411-2015     7031412-2014
 [29] 7031412-2015     7031413-2014     7031413-2015     7031414-2014
 [33] 7031414-2015     7031415-2014     7031415-2015     7031416-2015
 [37] 7031417-2015     7031418-2015     7031419-2015     7031420-2015
 [41] 7031608-2015     7031708-2015     7031815-2014     7042007-2015
 [45] 7042120-2014     7042515-2015     7042710-2014     7043108-2014
 [49] 7043120-2014     7043407-2015     7043618-2015     7053903-2015
 [53] 7053907-2015     7053916-2015     7054116-2015     7054119-2015
 [57] 7054207-2015     7054316-2014     7054319-2014     7054403-2015
 [61] 7054503-2014     7054504-2014     7054516-2015     7054520-2014
 [65] 7064602-2014     7064617-2014     7065006-2014     7065202-2015
 [69] 7065205-2015     7065215-2014     7065302-2015     7065312-2015
 [73] 7065318-2014     7085904-2015     7096216-2014     7096308-2015
 [77] 7096310-2015     7096315-2015     7096316-2015     7106402-2015
 [81] 7126920-2014     7127116-2014     7133004-2014     7133510-2015
 [85] 7133513-2014     7133515-2015     7137218-2015     7137219-2015
 [89] 7137317-2014     7137317-2017     8148207-2015     8148220-2015
 [93] 8159216-2014     8169515-2014     81710112-2014    81710203-2014
 [97] 81710504-2014    81710513-2014    81710517-2014    9807031614-2015
[101] 9807042103-2014  9807043618-2014  9807054316-2014  9807065212-2015
[105] 9807133512-2014  9808148207-2012  9808148220-2012  98081710316-2014
[109] 98081710317-2014 99070212018-2014 99070511013-2015 99070911620-2014
[113] 99071010814-2014 9908148515-2015  99081711213-2015 99081912406-2014
```

[2017-11-14 Abu email]:

- (T)hese 26 households are errors from double entry process. Please drop these IDs from these sections. The ID file is correct. → Not dealt with.

- There are two shamitee with the same gid=70314. After the baseline we followed only the group starting with 99. Thus we changed the gid to 9970314 (manually). Please kindly convert all the gid 70314 with 9970314. → Corrected.

Note: When I asked previously it was 26 HH-years but now 116 HH-years after corrections in other parts.

```
xid2[, gid := as.numeric(gid)]
xid2[gid == 70314, c("gid", "hhid") := list(9970314, as.integer(paste0(99, hhid)))]
xid2[, gid := factor(gid)]
```

A direct consequence of not having matching ID file is lacking treatment assignment information. As a patch, copy treatment assignment and group information across rounds in section files (but keep ID file uncorrected, as we expect its update later).

```
Z[-grep("id", names(Z))] ← lapply(Z[-grep("id", names(Z))],
        function(x) if (any(is.na(x[, Mgroup])))
        x[hhid %in% x[is.na(Mgroup), hhid],
        Mgroup := Mgroup[!is.na(Mgroup)][1], by = hhid] else x)
```

VI.3 Missing baseline

Drop these 116 individuals by dropping entries with Mgroup==NA. Below gives the number of rows being dropped, number of variables with all-NAs and not all-NAs (which indicate how much information is thrown away by this). Thanks to copying of Mgroup and other group information, the number of rows dropped is small.

	filenames	rowsDropped	allNAcols	not.allNAcols
1:	roster	0	58	0
2:	education	0	82	0
3:	contacts_with_mainland	0	66	0
4:	credit_and_borrowing	0	96	0
5:	input_output	0	137	0
6:	farm_production	3	24	17
7:	flood_related_information	0	59	0
8:	hh_asset	0	41	0
9:	hh_consumption	1	62	58
10:	id_updated_received_from_abu	0	0	0
11:	old_id_updated	0	0	0
12:	labor_income	0	88	0
13:	abu_livestockownershipupdated	2	48	18
14:	livestock_production	0	35	0
15:	monga	0	48	0
16:	ownership_and_contract	0	55	0
17:	poverty_updated	1	23	30
18:	poverty	1	21	22
19:	productive_assets	0	59	0
20:	relocation	0	55	0
21:	assistance	0	82	0
22:	savings_and_lending	0	47	0
23:	shocks	0	74	0
24:	women_empowerment	0	53	0
25:	MergedAssets	0	68	0
26:	LivestockLong	6	26	19
	filenames	rowsDropped	allNAcols	not.allNAcols

```
Z ← lapply(Z, function(x)
        if (any(grepl("Mg", colnames(x)))) x[!is.na(Mgroup),] else x)
```

HHs with no baseline: HHs whose disbursement was prior to their first interview.

```
HHnobaseline ← unique(xid2[survey == 1 & Disbursed1, hhid])
# xid3 is ID list of individuals with baseline info, xid2 includes individuals without bas
xid3 ← xid2[!(hhid %in% HHnobaseline), ]
```

If we drop individuals without baseline, it further reduces sample size by 93, and its breakdown of Mgroup, year is given in the below. They are all new group or replacing members.

```
      Mgroup
year   new group replacements
  2013        52           44
  2014        51           43
  2015        52           43
  2017        49           40
```

Below is the data list object we use in impact estimation.

```
ZB ← lapply(Z, function(x) x[!(hhid %in% HHnobaseline), ])
```

Save files. This is going to be used in the data preparation section of impact evaluation file.

```
saveRDS(Z, paste0(path1234, "data_read_in_a_list_with_treatment_patched.rds"))
saveRDS(ZB, paste0(path1234, "data_read_in_a_list_with_baseline_patched.rds"))
```

In what follows, all analysis is based on the sample with baseline.

VI.4 Panel structure by page

Names of sections in ./clean_panel_data_by_section/: roster, education, contacts_with_mainland, credit_and_borrowing, input_output, farm_production, flood_related_information, hh_asset, hh_consumption, id_updated_received_from_abu, old_id_updated, labor_income, abu_livestockownershipupdated, livestock_production, monga, ownership_and_contract, poverty_updated, poverty, productive_assets, relocation, assistance, savings_and_lending, shocks, women_empowerment, MergedAssets, LivestockLong

Names of sections in ./only_panel_2_3_4/: risk_pref_13, donations, 15, 21_2_income_generating_activities, 23_1, 23_2, 23_3, 23_4, 23_5, 24, behavioural_changes, s18_satisfaction_and_product_use, s19_q1_network_and_group_coordination, s19_q2_network_and_group_coordination, s19_q3_network_and_group_coordination, s19_q4_network_and_group_coordination, s19_q5_network_and_group_coordination, s19_q6_network_and_group_coordination, s21a_project_cycle, s21aprojectcycle, s21b_project_cycle, s21bprojectcycle, s22_q1-9_group_norms_and_leader, s22_q10-13_group_norms_and_leader, physical_asset, pre_caution, borrowing_2, by_product, dwelling_conditions, remittance, satisfaction, self_employed_income

```
Z.2 ← readRDS(paste0(path1234, "data_read_in_a_list_234.rds"))
jds ← fread(paste0(pathreceived, "DataForJDS.prn"))
```

Names of sections in ./raw_source_files/P1_Check_20170513, ./raw_source_files/P2_Check_20170513, ./raw_source_files/P3_Check_20170513, ./raw_source_files/P4_Check_20170513: s1_p1_2012_13, s1_1_p2, s1_2_p2, 1_houdehold_composition_2, 1_household_composition_1, s1.1, s1.2

```
Z ← readRDS(paste0(path1234, "data_read_in_a_list_with_baseline_patched.rds"))
xid ← readRDS(paste0(path1234, "ID.rds"))
jds ← fread(paste0(pathreceived, "DataForJDS.prn"))
# define o800
Z ← lapply(Z, function(x) {
  x[, o800 := 0L]
  x[hhid %in% jds[grepl("trea", treat), hhid], o800 := 1L]
  })
Z3new ← Z[[grep("roster", names(Z))]]
```

Below tabulation shows many unmatched hhid across rounds in roster. (FALSE indicates no match in other rds.)

```
   year TRUE FALSE
1: 2012 1506    94
2: 2013  516     7
3: 2014 1983     0
4: 2015 1994     0
5: 2017 1914     0
```

Original 800 HHs not found in other rounds of roster files. (FALSE indicates no match in other rds.)

```
   year TRUE FALSE
1: 2012  759    41
2: 2013    0     0
3: 2014  743     0
4: 2015  745     0
5: 2017  708     0
```

101 HHs in 2012/2013 with unmatched hhid in subsequent rds in roster files: 7010103, 7010104, 7010113, 7020217, 7020219, 7020313, 7020315, 7020501, 7020502, 7020503, 7020504, 7020505, 7020506, 7020507, 7020508, 7020509, 7020510, 7020511, 7020512, 7020513, 7020514, 7020515, 7020516, 7020517, 7020518, 7020519, 7020520, 7021116, 7021210, 7031401, 7031402, 7031403, 7031404, 7031405, 7031406, 7031407, 7031408, 7031409, 7031410, 7031411, 7031412, 7031413, 7031414, 7031415, 7031416, 7031417, 7031418, 7031419, 7031420, 7031502, 7031505, 7031513, 7031602, 7031608, 7031612, 7042013, 7042103, 7043407, 7053909, 7054104, 7054106, 7054408, 7054413, 7054416, 7054419, 7054502, 7054516, 7064603, 7064604, 7065313, 7075702, 7085901, 7096206, 7126813, 7133504, 7133512, 7133514, 7133516, 7133520, 7137304, 7137310, 7137317, 7137320, 8147811, 8147903, 8148013, 8148207, 8148220, 8158816, 8159220, 8169615, 8169719, 8169815, 81710316, 9807031614, 9808169612, 9907031415, 99070211912, 99081412508, 99081412509, 99081711213. Among which 68 are dropped out HHs. Below tabulation shows there are 27 cases of continuing members not being captured after 2012. Although classified as continuing members, are they drop outs? (Remaining 6 cases?)

o800: 41 HHs in 2012/2013 with unmatched hhid in subsequent rds in roster files: 7010103, 7010104, 7020313, 7020507, 7020508, 7020509, 7020510, 7020512, 7020513, 7020514, 7020516, 7020519, 7020520, 7031401, 7031402, 7031403, 7031404, 7031406, 7031408, 7031410, 7031411, 7031418, 7031419, 7031502, 7031505, 7031513, 7053909, 7054106, 7054408, 7054413, 7054516, 7065313, 7075702, 7096206, 7133504, 7137310, 8147811, 8148013, 8148207, 8158816, 8169615. Among which 33 are dropped out HHs. Below tabulation shows there are 8 cases of continuing members not being captured after 2012. Although classified as continuing members, are they drop outs? (Remaining 0 cases?)

[2017-11-14 Abu email]: (T)hese households took the loan but have migrated to Dhaka or other places and could not be traced. → Create RanAway = T/F.

```
                  Mstatus
Assign             gErosion gRejection iRejection iReplacement newGroup
  traditional             0          0          0            0        0
  large                   0          0          0            0        0
  large grace             0          0          0            0        0
  cow                     0          0          0            0        0
  drop outs               0         22         22            0        0
  forced drop outs       24          0          0            0        0
                  Mstatus
Assign             oldMember
  traditional              2
  large                   10
  large grace              4
  cow                     11
  drop outs                0
  forced drop outs         0
```

```
      gid    hhid   memname year   Mstatus      Assign creditstatus
 1: 70101 7010103   khoteza 2012 oldMember       large          Yes
```

```
 2: 70101 7010104    rupali 2012 oldMember       large          Yes
 3: 70101 7010113     rotna 2012 oldMember       large          Yes
 4: 70202 7020217    suroti 2012 oldMember         cow          Yes
 5: 70202 7020219    halima 2012 oldMember         cow          Yes
 6: 70203 7020313   morjina 2012 oldMember       large          Yes
 7: 70203 7020315    rokeya 2012 oldMember       large          Yes
 8: 70211 7021116     rajia 2012 oldMember traditional          Yes
 9: 70212 7021210   ronjona 2012 oldMember         cow          Yes
10: 70315 7031513    rahima 2012 oldMember traditional          Yes
11: 70316 7031608     omisa 2012 oldMember         cow          Yes
12: 70420 7042013    sahena 2012 oldMember       large          Yes
13: 70434 7043407    hajera 2012 oldMember       large          Yes
14: 70541 7054104   aynaful 2012 oldMember         cow          Yes
15: 70545 7054516 shahera 2 2012 oldMember large grace          Yes
16: 70646 7064603    saleha 2012 oldMember large grace          Yes
17: 70859 7085901     safia 2012 oldMember         cow          Yes
18: 70962 7096206 sada rani 2012 oldMember       large          Yes
19: 71268 7126813       nur 2012 oldMember       large          Yes
20: 71373 7137304   lalbuni 2012 oldMember         cow          Yes
21: 71373 7137310   shahida 2012 oldMember         cow          Yes
22: 71373 7137317    afruja 2012 oldMember         cow          Yes
23: 71373 7137317    afruja 2015 oldMember         cow          Yes
24: 71373 7137320  monoyara 2012 oldMember         cow          Yes
25: 81592 8159220    sirina 2012 oldMember large grace          Yes
26: 81696 8169615    rahela 2012 oldMember       large          Yes
27: 81698 8169815     azifa 2012 oldMember large grace          Yes
      gid    hhid   memname year   Mstatus      Assign creditstatus
```

```
xid2[, RanAway := F]
xid2[hhid %in% attritedHH & grepl("^old", Mstatus), RanAway := T]
```

VII Plots

```
Z ← readRDS(paste0(path1234, "data_read_in_a_list_with_baseline_patched.rds"))
```

VII.1 Incomes

Revenues are reported partially.

```
      hhid        year                    Mgroup                    Assign
 7020308:  4   2012: 22   continued        :328   traditional      :104
 7020902:  4   2013:  2   drop outs        : 29   large            :177
 7021216:  4   2014:192   forced drop outs:  5   large grace      :119
 7020301:  3   2015:186   new group        :156   cow              : 89
 7020408:  3   2017:123   replacements     :  7   drop outs        : 29
 7020604:  3                                     forced drop outs:  5
 (Other):504                                     NA's             :  2
  TotalRevenue     Panel
 Min.   :   700   1: 24
 1st Qu.: 10400   2:192
 Median : 18000   3:186
 Mean   : 20961   4:123
 3rd Qu.: 26800
 Max.   :399800
```

Costs are reported partially. There are 22, 2, 192, 186, 123 HHs who report revenues for 2012, 2013, 2014, 2015, 2017, only 15, 3, 1 HHs report costs for 2012, 2014, 2015, respectively.

Table 2: Files and survey rounds

filename	rds	2012	2013	2014	2015	2017
s1 (roster)	4	6729	2149	8625	8590	7956
education	4	6725	2059	2799	2719	2640
contacts_with_mainland	3	1595	485	1980	1994	
credit_and_borrowing	4	1597	502	1982	1995	1913
input_output	4	1597	504	1982	1995	1915
farm_production	4	22	2	211	223	131
flood_related_information	3	1595	500	1980	1993	
hh_asset	4	926	399	4057	6567	7162
hh_consumption	3			1980	1994	1914
id_updated_received_from_abu	4	1600	523	1983	1995	1914
old_id_updated	4	1600	523	1983	1995	1914
labor_income	4	6725	2057	2398	2463	2361
abu_livestockownershipupdated	4	1595	504	1741	1939	1837
livestock_production	4	3190	1004	3958	3986	3826
monga	4	1595	502	1979	1994	1913
ownership_and_contract	4	1595	502	2022	2021	1913
poverty_updated	3			2397	2461	2360
poverty	3			1980	1994	1914
productive_assets	4	1472	496	1965	1990	1898
relocation	3	1595	485	1980	1994	
assistance	4	1595	501	1974	1993	1913
savings_and_lending	4	1594	502	1980	1994	1913
shocks	1	1597	504			1915
women_empowerment	4	1594	499	1978	1992	1911
MergedAssets	4	1486	500	1974	1994	1915
LivestockLong	4	4785	1512	5223	5817	5511
borrowing_2	3			2083	2094	2004
by_product	3			2164	2133	2014
dwelling_conditions	3			2083	2094	2004
remittance	3			2083	2094	2004
satisfaction	3			2083	2094	2004
self_employed_income	3			2097	2106	2011
risk_pref_13	1			2083		
donations	1				2094	

Source: Compiled from GUK data.

Notes: 1. Number of rows are displayed.

2. 2012 and 2013 are round 1. 2012 and 2013 data were jointly reported for s1 (roster) but separated by using information from id file. MergedAssets is a merged file of hh_asset and productive_assets.

```
      hhid          year      TotalOfCosts   Panel
7020306: 1   2012:15   Min.    :  70   1:15
7020308: 1   2014: 3   1st Qu.: 445   2: 3
7020319: 1   2015: 1   Median : 754   3: 1
7020902: 1             Mean   : 848
7021208: 1             3rd Qu.: 995
7021216: 1             Max.   :2255
(Other):13
```

Plot agricultural revenues (Figure 13). Check trends in HH total labour income (Figure 14, 15).

Table 3: Files and survey rounds for original 800 HHs

filename	rds	2012	2013	2014	2015	2017
s1 (roster)	4	3353		3230	3204	2924
education	4	3340		1040	1019	984
contacts_with_mainland	2	796		742	745	
credit_and_borrowing	4	796		742	744	708
input_output	4	796		742	745	708
farm_production	4	12		65	65	43
flood_related_information	2	796		742	744	
hh_asset	4	450		1481	2423	2602
hh_consumption	3			741	744	707
id_updated_received_from_abu	4	800		743	745	708
old_id_updated	4	800		743	745	708
labor_income	4	3340		884	911	866
abu_livestockownershipupdated	4	796		666	730	676
livestock_production	4	1592		1484	1490	1416
monga	4	796		742	745	708
ownership_and_contract	4	796		759	759	708
poverty_updated	3			883	910	865
poverty	3			741	744	707
productive_assets	4	734		735	742	703
relocation	2	796		742	745	
assistance	4	796		750	745	708
savings_and_lending	4	796		742	745	708
shocks	0	796				708
women_empowerment	4	795		741	744	707
MergedAssets	4	741		738	744	708
LivestockLong	4	2388		1998	2190	2028
borrowing_2	3			742	746	708
by_product	3			768	759	710
dwelling_conditions	3			742	746	708
remittance	3			743	746	708
satisfaction	3			742	746	708
self_employed_income	3			750	750	710
risk_pref_13	1			743		
donations	1				746	

Source: Compiled from GUK data.

Notes: 1. Number of rows are displayed.

2. 2012 and 2013 are round 1. 2012 and 2013 data were jointly reported for s1 (roster) but separated by using information from id file. MergedAssets is a merged file of hh_asset and productive_assets.

VII.2 Assets

Plot asset values (Figure 16), asset earning (Figure 17), and newly purchased asset values asset values (Figure 18).

Livestock asset values are given in Figure 19. Livestock produce sales using imputed prices are given in Figure 20.

Table 4: Files and survey rounds for original 800 HHs (continued)

filename	rds	2012	2013	2014	2015	2017
15	1				746	
21_2_income_generating_activities	1				1133	
23_1	1				745	
23_2	1				901	
23_3	1				751	
23_4	1				5968	
23_5	1				5968	
24	1				746	
behavioural_changes	1					708
s18_satisfaction_and_product_use	1					708
s19_q1_network_and_group_coordination						3540
s19_q2_network_and_group_coordi	1					3524
s19_q3_network_and_group_coordination						790
s19_q4_network_and_group_coordi	1					722
s19_q5_network_and_group_coordination						926
s19_q6_network_and_group_coordi	1					722
s21a_project_cycle	1					832
s21aprojectcycle	1					832
s21b_project_cycle	1					1397
s21bprojectcycle	1					1397
s22_q1-9_group_norms_and_leader	1					708
s22_q10-13_group_norms_and_leader	1					708

Source: Compiled from GUK data.

Notes: 1. Number of rows are displayed.

2. 2012 and 2013 are round 1. 2012 and 2013 data were jointly reported for s1 (roster) but separated by using information from id file. MergedAssets is a merged file of hh_asset and productive_assets.

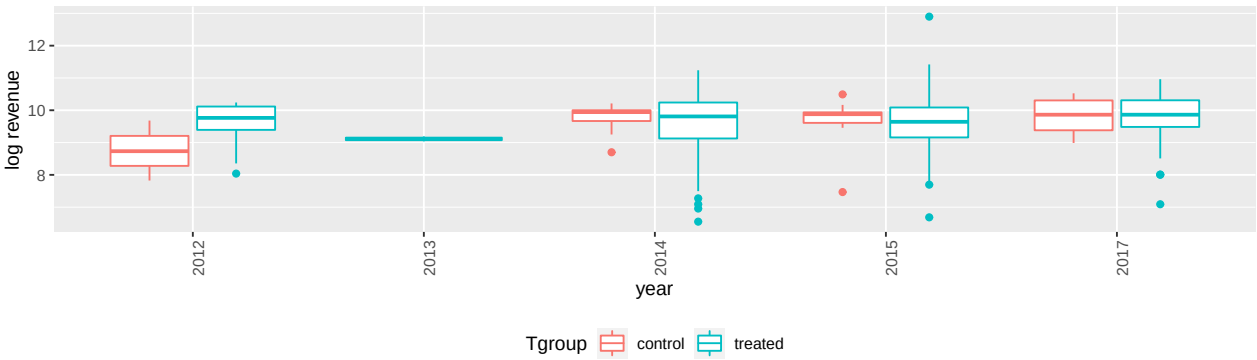

Figure 13 Farming revenues

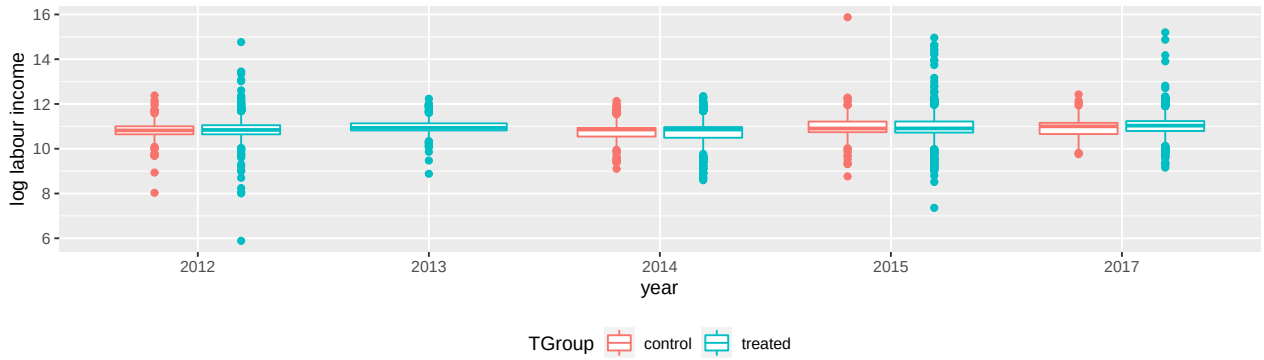

Figure 14 Labour incomes

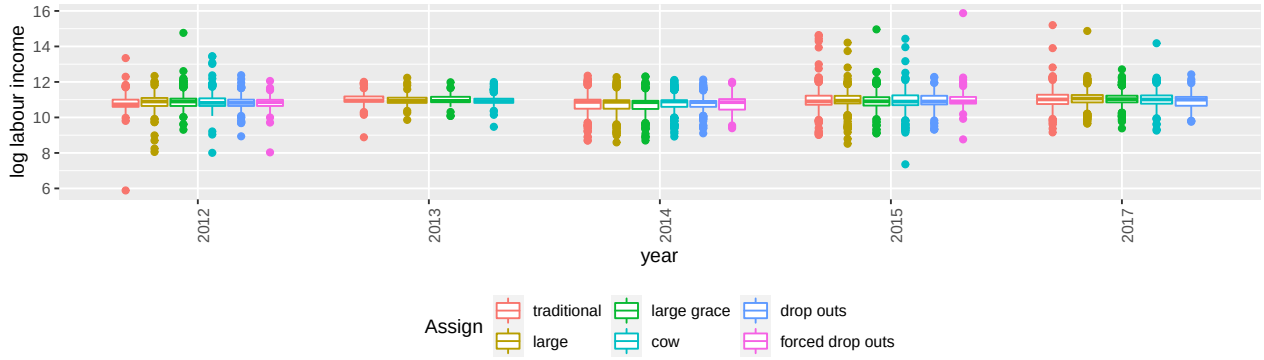

Figure 15 Labour incomes bewteen arms and controls

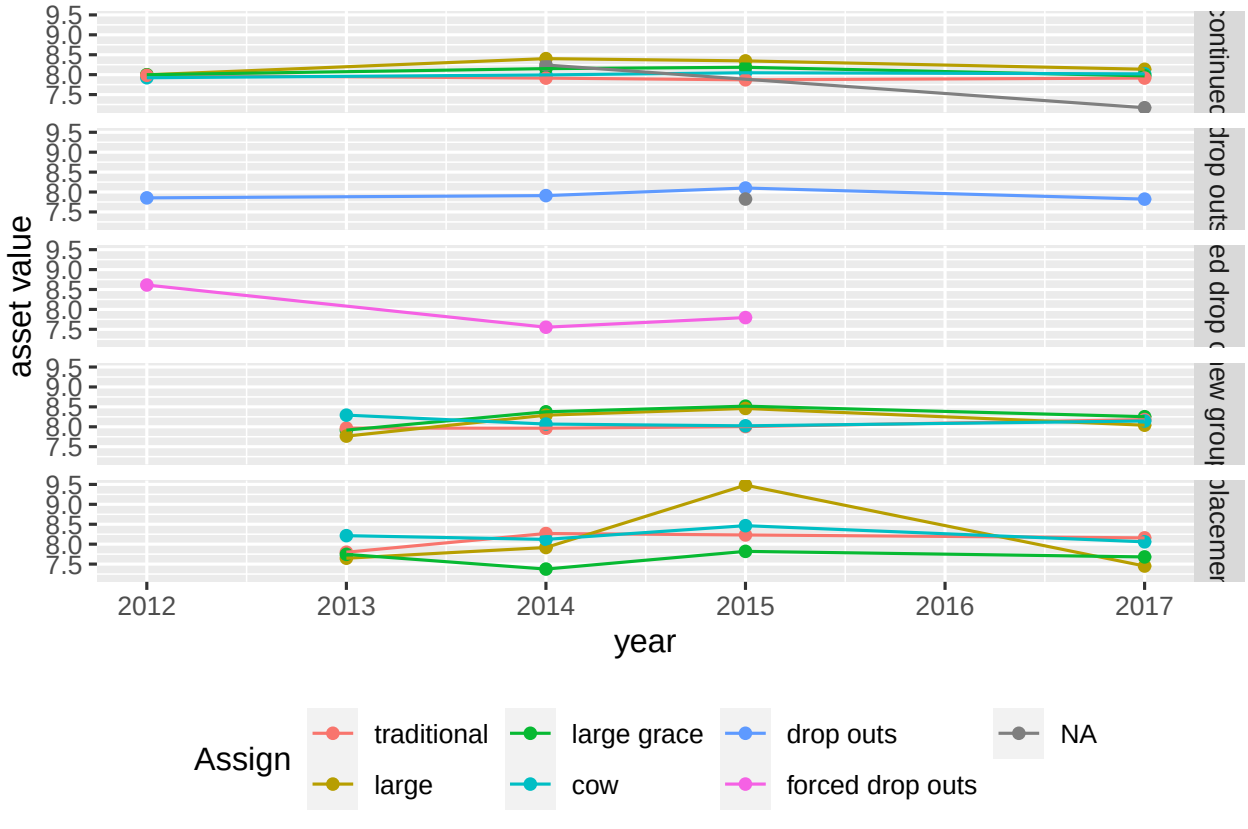

Figure 16 Asset value by arms

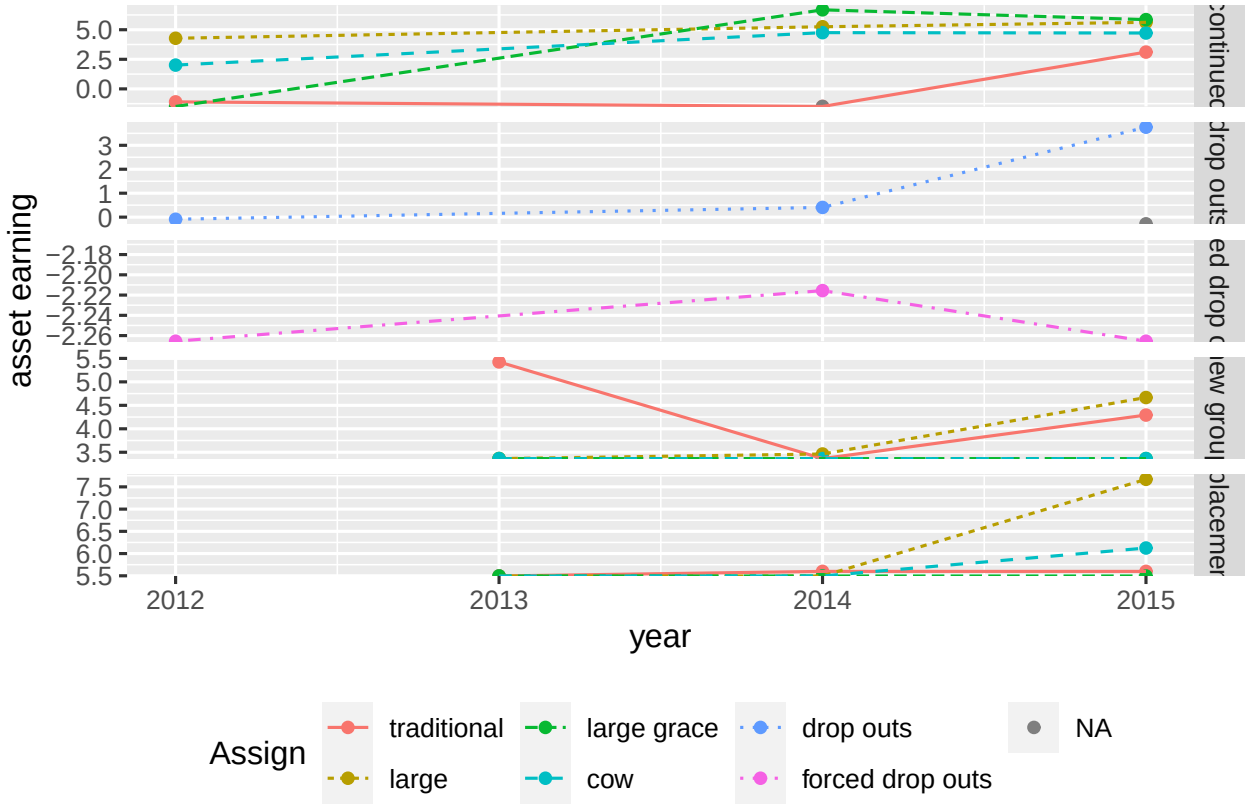

Figure 17 Asset earnings by arms

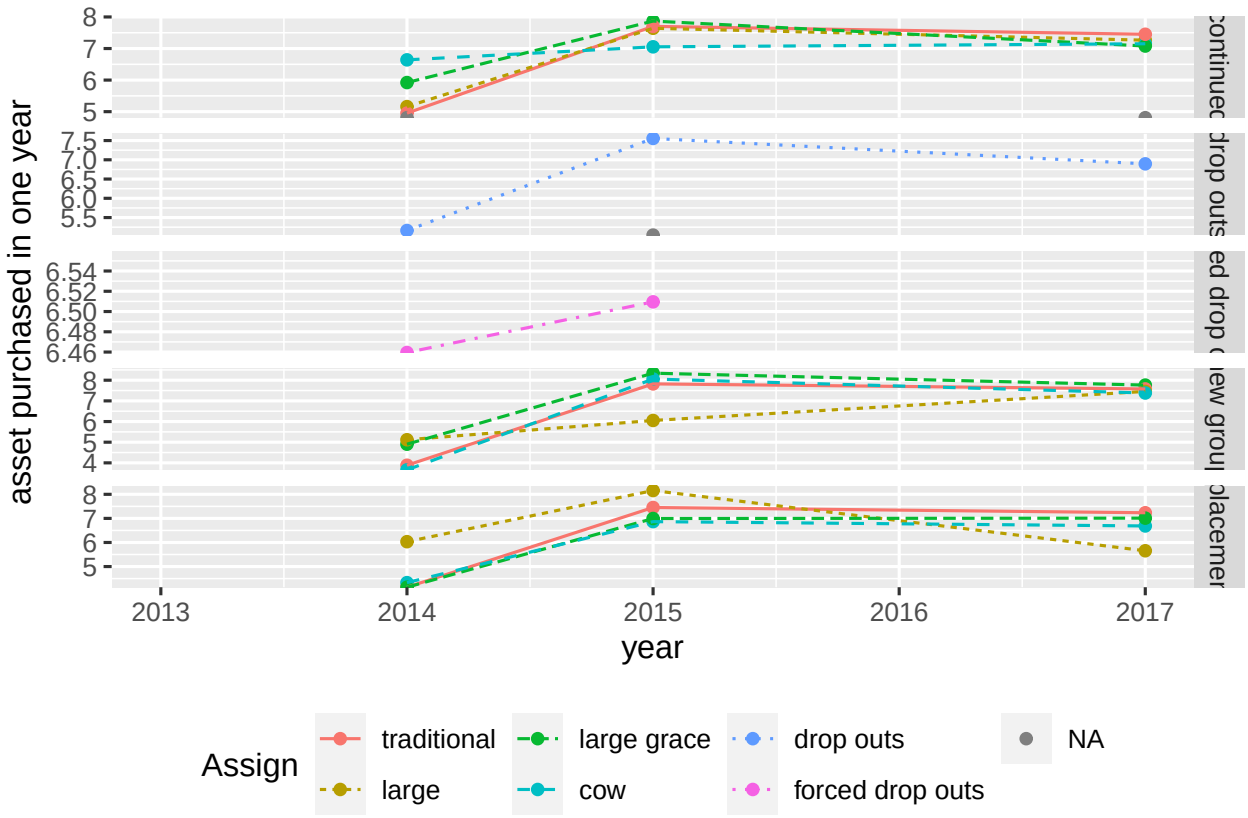

Figure 18 Asset purchased in last one year by arms

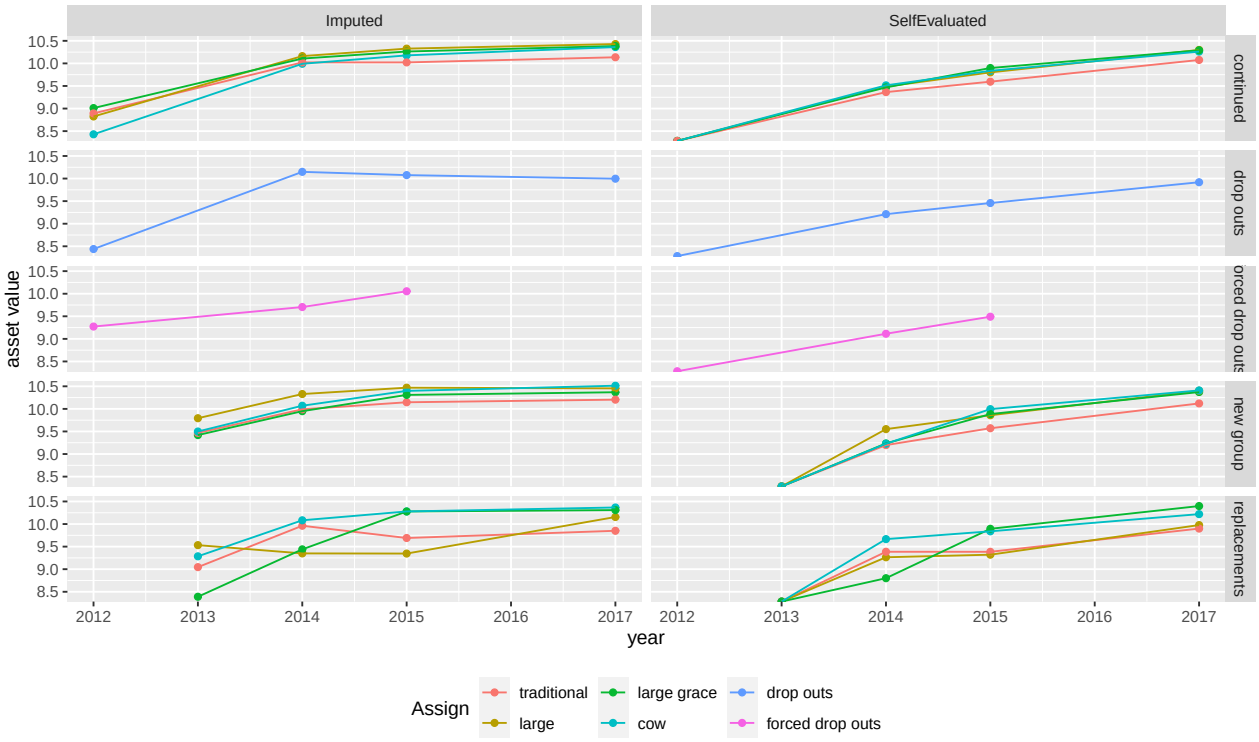

Figure 19 Livestock asset value by arms

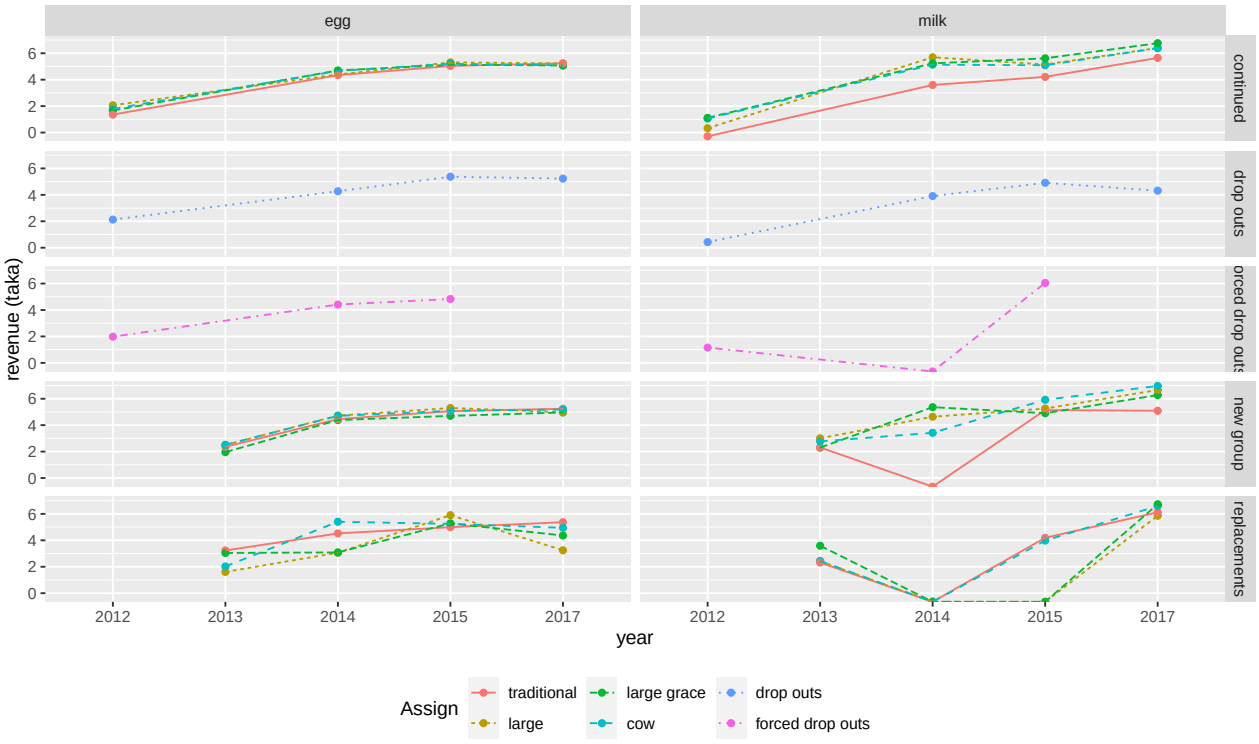

Figure 20 Livestock produce sales by arms

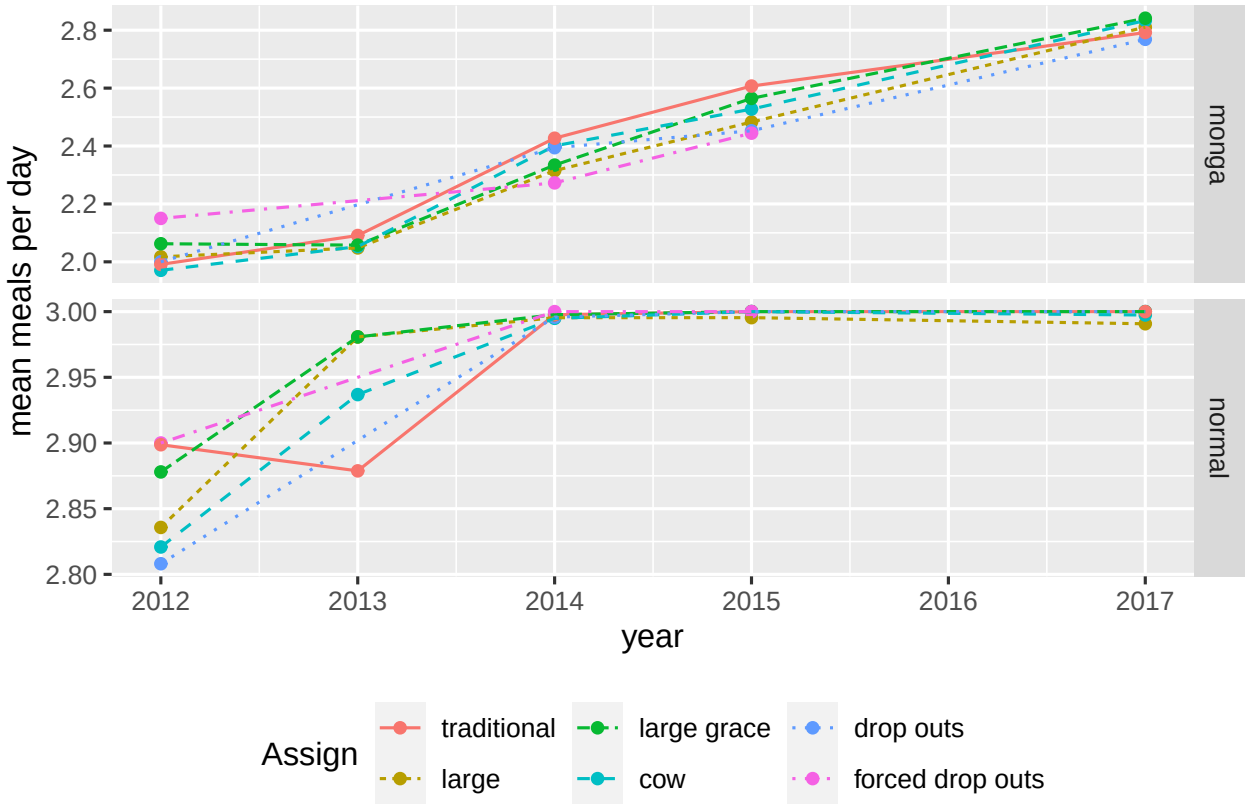

Figure 21 Meals per day by arms and controls

VII.3 Poverty

VII.4 Schooling

	AgeComputed													
Enrolled	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	416	347	222	125	72	91	49	133	84	107	173	158	136	158
1	113	312	695	843	879	1166	807	1016	791	581	643	325	337	221

- (Voluntary) drop out group has lower enrollment rates.

Plot enrollment (Figure 24).

Figure 22 Saving by arms and controls

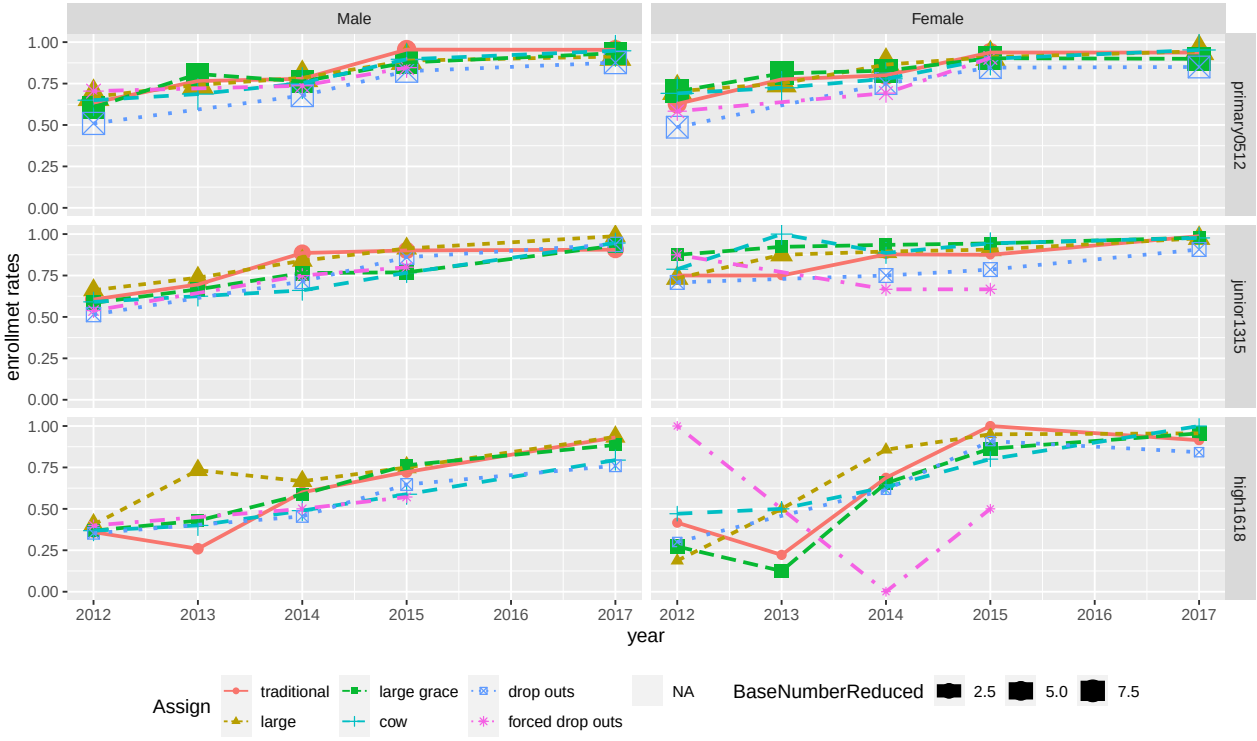

Figure 23 School enrollment by arms and controls

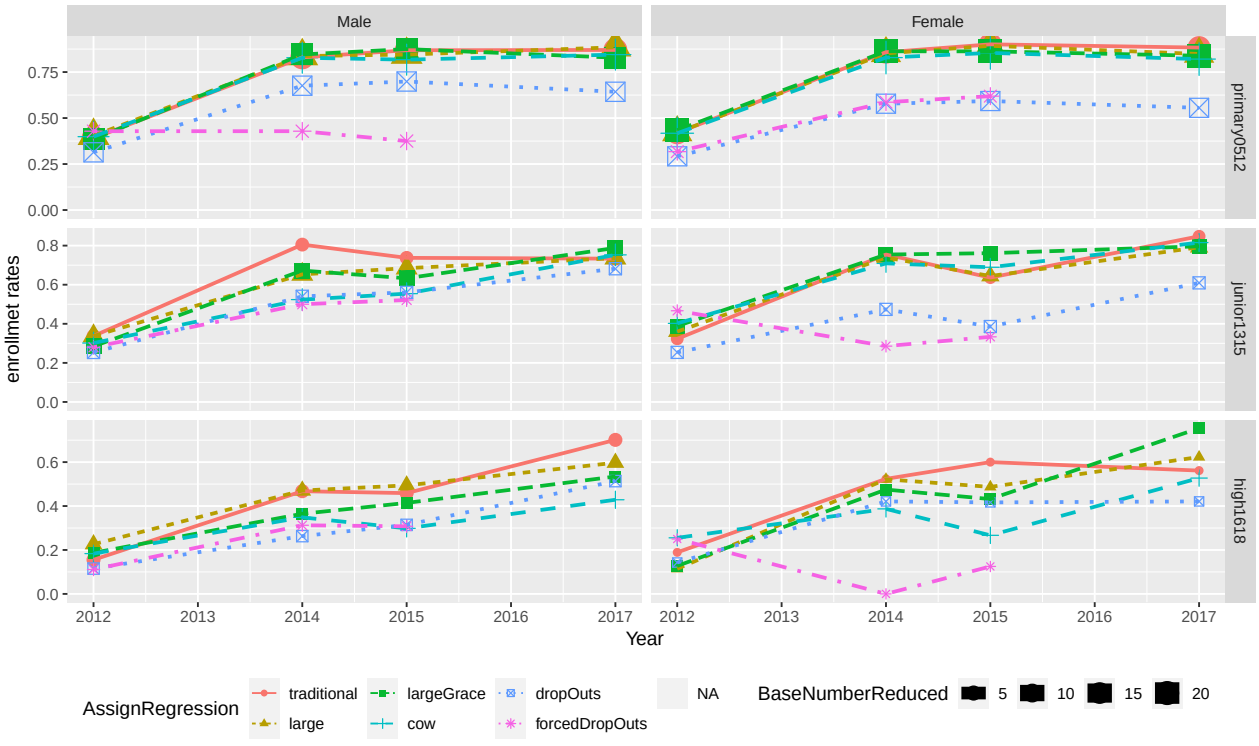

Figure 24 School enrollment with artificially augmented panel by arms and controls