Comparing outcomes between groups

March 15, 2017
10:05

Seiro Ito

Contents

There are a few key variables.

- receivedCredit: The actual treatment status. This is time-invariant. It is T if a subject is receives a loan in our observation period.
- assignment: The original treatment assignement. This is time-invariant. This differs from receivedCredit because in our design everyone is deemed to get treated at the end (but there are subjects who opted out of a loan but remained in a group).
- disbursed: If a subject received a loan. This is time-variant.
- elapsed: The number of days since receiving a loan at rd 3 interview date. This is time-invariant as it is computed only at rd 3. This defines the eventual treatment dose and should be our main covariate.

Following results are obtained.

Figure 1 This plots the mean of "3 meals per day" in each round. Left panel is control vs. treated in receivedCredit (actual assignmemt). Right panel is control vs. treated in assignment (original assignment). The question is changed in rd 2 onwards so direct comparison across rd 1 and 2,3 are not tenable. However, one sees a rising "3 meals per day", but almost paralel trend between rd 2 and 3 (which are comparable).

Table 1 This is a first-difference (linear probability) estimation result of "3 meals per day" on disbursed, arm, their interactions, assignment, elapsed days, using the post treatment data of rds 2, 3. It shows positive impacts of receiving a loan (disbursed under the covariate name "credit") aafter controlling for arm. control/treated are relative to "lost to flood" or "rejections," so it is not surprising to have better food intake.

Figure 2 Livestock is the main stated usage of loans.

Figure 3 Work hours seem to get longer.

Figure 4 New loans increased in rd 2, whose recall period corresponds to the timing disbursement.

Figure 5 Asset holding by receivedCredit= T/F and rds = 1,2,3. Asset holding is computed with rd 1 asset holding, asset addition in rd 2, 3, while assuming an annual rate of 5% depreciation. receivedCredit not randomised allocation as loan receipt must be agreed by subjects. It shows that the loan receivers have higher mean asset in rd 3, but not in rd 1 or 2, where the latter is good. Red dotted lines are medians, blue dotted lines are means.

Figure 6 Asset holding by disbursed=T/F and rds=1,2,3. This is also an endogenous switch. The basic picture is the same as Figure 5.

Figure 7 Asset holding by elapsed days grouped into "early receivers" and "late receivers" according to median elapsed day. This is (roughly) a randomised switch. It shows increasing asset levels, but no mean or median difference between early and late receivers.

Figure 8 Asset holding by elapsed days and arms. Not much to see here.

Figure 9 Difference in group-average asset holding between the treated and the control in assignment by difference in elapsed and arms. So it is group differences among original treatment assignment (treated - control) within the same cluster. It controls for cluster FE, and dose and outcome differences are taken between randomly assigned treatment status. This should be one of our main comparisons. (This is not DID: If I plot the first-difference version of the plots between rds, it will be double difference estimates.) When I draw loess curves, I see no trend over various elapsed day ("treatment dose") differences. It hints a zero gradient of dose levels.

Figure 10 Same identification idea as Figure 9 on livestock values. Again, we do not see markedly strong impacts, but we see some differences in terms of dispersion. cow arm has smaller variations around the loess curves in rds 2 and 3 relative to the large grace arm. traditional arm has a simlar pattern as large grace.

Figure 11 Livestock holding by elapsed days. Substantial heterogeneity in rd 3 but no statistical significant changes.

Figure 13 Total asset holding by elapsed days. Substantial heterogeneity in rd 3 but no statistical significant changes.

Table 3 DID estimation of total asset holding by elapsed days. Zero impact.

I Read

List folder names and read files.

```
setwd(pathsource.mar)
foldername ← list.dirs(path = ".", recursive = T, full.names = T)
foldername ← foldername[!grepl("add|ori|^\\.$|1$", foldername)]
fn ← unique(list.files(path = foldername, pattern = ".prn$",
      recursive = T, full.names = T))
X = lapply(fn, fread, integer64 = "double")
```

II Treatment through time

Treatment assignment file. There are 220 cases of attrition who are group rejections (140) and lost to flood (80).

```
setwd(pathsave)
```

```r
tr <- fread("treatment_assignment.prn")
tr[, disburseDate := as.POSIXct(disburseDate, format = "%Y-%m-%d")]
tr[, purchaseDate := as.POSIXct(purchaseDate, format = "%Y-%m-%d")]
tr[, arm := factor(arm)]
tr[, memstatus := factor(memstatus)]
tr[, assignment := factor(assignment)]
tr0 <- tr[, .(gid, hhid, memstatus, assignment, arm, receivedCredit, elapsed, daysFromSta
table0(tr0[, .(memstatus, assignment)])
```

```
                      assignment
memstatus                     control treated
  group rejection        140       0       0
  individual rejection     0      69      90
  lost to flood           80       0       0
  new group                0     230     230
  old                      0     620     599
  replacement              0      69      90
```

```r
table0(tr0[, .(arm, assignment)])
```

```
                     assignment
arm                          control treated
  before intervention  46       0       0
  cow                   0     254     256
  large                40     233     239
  large grace           0     239     243
  lost to flood        56       0       0
  traditional          78     262     271
```

```r
table0(tr0[, .(memstatus, arm)])
```

```
                      arm
memstatus               before intervention cow large large grace lost to flood
  group rejection                        22   0    40           0             0
  individual rejection                    0  72    12          22             0
  lost to flood                          24   0     0           0            56
  new group                               0  60   100         100             0
  old                                     0 306   348         338             0
  replacement                             0  72    12          22             0
                      arm
memstatus               traditional
  group rejection                78
  individual rejection           53
  lost to flood                   0
  new group                     200
  old                           227
  replacement                    53
```

```r
tr1 <- tr0[, -grep("Cre", colnames(tr0)), with = F]
```

tr0 and tr1 are based on the information at rd 3.

```r
setwd(pathsave)
indate <- fread("interview_dates_long.prn", integer64 = "double")
indate[, intDate := as.POSIXct(intDate, format = "%Y-%m-%d")]
indate[, daysSince2014 :=
        asn(intDate - as.POSIXct("2014-01-01", format = "%Y-%m-%d"))]
indate <- reshape(indate, direction = "wide", idvar = "hhid",
        timevar = "rd",
        v.names = c("disbursed", "purchased", "intDate", "daysSince2014"))
```

Merge interview dates with treatment assignment info tr1.

```r
setkey(indate, hhid); setkey(tr0, hhid); setkey(tr1, hhid)
tr0 ← indate[tr0]
tr1 ← indate[tr1]
```

```r
tr1l ← reshape(tr1, direction = "long", idvar = c("gid", "hhid",
        "assignment", "arm", "memstatus"), varying = grepout("\\.\\d", colnames(tr0)))
setnames(tr1l, "time", "rd"); setkey(tr1l, hhid, rd)
table(tr1l[, .(rd, disbursed, assignment)], useNA = "ifany")
```

```
, , assignment = 

   disbursed
rd  FALSE TRUE <NA>
  1   220    0    0
  2   173    0   47
  3   168    0   52

, , assignment = control

   disbursed
rd  FALSE TRUE <NA>
  1   988    0    0
  2   390  409  189
  3   148  798   42

, , assignment = treated

   disbursed
rd  FALSE TRUE <NA>
  1  1009    0    0
  2   127  770  112
  3   103  871   35
```

```r
table(tr1l[, .(rd, disbursed, memstatus)], useNA = "ifany")
```

```
, , memstatus = group rejection

   disbursed
rd  FALSE TRUE <NA>
  1   140    0    0
  2   118    0   22
  3   114    0   26

, , memstatus = individual rejection

   disbursed
rd  FALSE TRUE <NA>
  1   159    0    0
  2   124    0   35
  3   130    0   29

, , memstatus = lost to flood

   disbursed
rd  FALSE TRUE <NA>
  1    80    0    0
  2    55    0   25
  3    54    0   26
```

```
, , memstatus = new group

   disbursed
rd  FALSE TRUE <NA>
  1   460    0    0
  2    90  332   38
  3    37  416    7

, , memstatus = old

   disbursed
rd  FALSE TRUE <NA>
  1  1219    0    0
  2   265  756  198
  3    64 1117   38

, , memstatus = replacement

   disbursed
rd  FALSE TRUE <NA>
  1   159    0    0
  2    38   91   30
  3    20  136    3
```

III Food consumption (23B in rd 1, 3B in rds 2, 3)

```
setwd(pathsource.mar)
grepout("sect.*\\_3b|23b_m|23b.prn", fn)
```

```
[1] "./1/combined/s23b.prn" "./2/section_3b.prn"    "./3/section_3b.prn"
```

```
sec3b = copy(X[grep("sect.*\\_3b|23b_m|23b.prn", fn)])
setnames(sec3b[[2]], "id", "hhid")
setnames(sec3b[[3]], "id", "hhid")
```

There is pure duplication in rd 1 files. Drop them.

```
lapply(sec3b[1],
        function(x) x[hhid %in% x[duplicated(x[, .(hhid, mid)]), hhid], ])
```

```
[[1]]
           hhid mid s23b_1 s23b_2 s23b_31_fish s23b_32_meat s23b_33_egg s23b_5
 1: 9808148207   1      3      2            1           NA           1      2
 2: 9808148207   1      3      3            1           NA           2      2
 3: 9808148207   2     NA     NA           NA           NA          NA     NA
 4: 9808148207   2     NA     NA           NA           NA          NA     NA
 5: 9808148207   3     NA     NA           NA           NA          NA     NA
 6: 9808148207   3     NA     NA           NA           NA          NA     NA
 7: 9808148207   4     NA     NA           NA           NA          NA     NA
 8: 9808148207   5     NA     NA           NA           NA          NA     NA
 9: 9808148207   6     NA     NA           NA           NA          NA     NA
10: 9808148207   7     NA     NA           NA           NA          NA     NA
11: 9808148220   1      3      2            1           NA           1      2
12: 9808148220   1      3      3            2           NA           2      2
13: 9808148220   2     NA     NA           NA           NA          NA     NA
14: 9808148220   2     NA     NA           NA           NA          NA     NA
15: 9808148220   3     NA     NA           NA           NA          NA     NA
```

```
16: 9808148220   3     NA     NA           NA           NA          NA     NA
17: 9808148220   4     NA     NA           NA           NA          NA     NA
18: 9808148220   5     NA     NA           NA           NA          NA     NA
    s23b_6 s23b_71_fish s23b_72_meat s23b_73_egg      u_id
 1:      1           99           NA          NA        NA
 2:      2            2           NA          NA 9808148480
 3:     NA           NA           NA          NA        NA
 4:     NA           NA           NA          NA 9808148480
 5:     NA           NA           NA          NA        NA
 6:     NA           NA           NA          NA 9808148480
 7:     NA           NA           NA          NA 9808148480
 8:     NA           NA           NA          NA 9808148480
 9:     NA           NA           NA          NA 9808148480
10:     NA           NA           NA          NA 9808148480
11:      1           99           NA          NA        NA
12:      2            1           NA          NA 9808148480
13:     NA           NA           NA          NA        NA
14:     NA           NA           NA          NA 9808148480
15:     NA           NA           NA          NA        NA
16:     NA           NA           NA          NA 9808148480
17:     NA           NA           NA          NA 9808148480
18:     NA           NA           NA          NA 9808148480
```

```
sec3b[1] ← lapply(sec3b[1],
        function(x) x[!duplicated(x[, .(hhid, mid)]), ])
sec3b[[2]] ← sec3b[[2]][!duplicated(hhid), ]
asl(lapply(sec3b[1],
        function(x) any(duplicated(x[, .(hhid, mid)]))))
```

```
[1] FALSE
```

There is only one HH that reports food intake of a non-head member. I will drop this non-head member.

```
table0(sec3b[[1]][!is.na(s23b_1), mid])
```

```

   1    4
2216    1
```

```
sec3b[[1]][hhid %in% hhid[!is.na(s23b_1) & mid != 1], ]
```

```
          hhid mid s23b_1 s23b_2 s23b_31_fish s23b_32_meat s23b_33_egg s23b_5
1: 99081412516   1      3      2            1           NA          NA      2
2: 99081412516   2     NA     NA           NA           NA          NA     NA
3: 99081412516   3     NA     NA           NA           NA          NA     NA
4: 99081412516   4      3      2            1           NA           1      2
   s23b_6 s23b_71_fish s23b_72_meat s23b_73_egg u_id
1:      1           99           NA          NA   NA
2:     NA           NA           NA          NA   NA
3:     NA           NA           NA          NA   NA
4:      2            1           NA          NA   NA
```

```
# drop all non-head members
sec3b[[1]] ← sec3b[[1]][mid == 1, ]
```

Drop hhid = NA.

```
sec3b ← lapply(sec3b, function(x) x[!is.na(hhid), ])
```

Merge treatment assignment info.

```
invisible(lapply(sec3b, setkey, hhid)); setkey(tr0, hhid)
sec3b ← lapply(sec3b, merge, tr0, by = "hhid", all.x = T)
```

Some gids are missing in sec3b. Check if the merge is done correctly. Check if this is due to hhid = 980... cases. Strip leading 980/990 and see if the matched observations have variables originally from tr0.

```
nahhids ← lapply(sec3b, function(x) asn(unique(x[is.na(gid), hhid])))
nahhids ← lapply(nahhids, gsub, pattern = "^980|^990", replacement = "")
nahhids ← lapply(nahhids, asn)
table0(sec3b[[1]][hhid %in% nahhids[[1]], assignment])
```

```

control treated
      7       3
```

```
table0(sec3b[[2]][hhid %in% nahhids[[2]], assignment])
```

```
<NA>
   2
```

```
table0(sec3b[[3]][hhid %in% nahhids[[3]], assignment])
```

```
<NA>
   4
```

Rd 1 seems to be merged OK. Rd 2, 3 show that there are duplicated hhid so drop all entries with duplication.

```
sec3b[[2]] ← sec3b[[2]][!(hhid %in% nahhids[[2]]), ]
sec3b[[3]] ← sec3b[[3]][!(hhid %in% nahhids[[3]]), ]
```

There still remains unmatched observations as seen in NAs in assignment (found in Sec 3B files but not in identification files.) We drop these observations.

```
lapply(sec3b, function(x) table0(x[is.na(gid), assignment]))
```

```
[[1]]
<NA>
  35

[[2]]
<NA>
   4

[[3]]
named integer(0)
```

```
sec3b ← lapply(sec3b, function(x) x[!is.na(gid), ])
asn(lapply(sec3b, dim))
```

```
[1] 2183   38 2072   59 2089   59
```

Three meals. In rd 1, we ask for all the members about the number of times they eat meals, during monga and off-monga seasons. On average, there is only 1 out of 1 HH members reponding to the question, which are all HH head members. In rds 2 and 3, we ask a blanket question if all the members eat three times a day for the whole year. So rd 1 question is more likely to be responded as “3 times” than in rd 2, 3 questions, *cetris paribus*. So observing more “3 times” responses in the latter rds indicate that there may be improvements in household food intake.

Combine rd 1 original and additional into a single file, then put into a list with rds 2, 3.

```
meal3.0 <- c(asn(table0(grepl("3", sec3b[[1]][, s23b_1]) & grepl("3", sec3b[[1]][, s23b_2
         grepl("3", sec3b[[1]][, s23b_5]) & grepl("3", sec3b[[1]][, s23b_6]))),
         asn(lapply(sec3b[2:3], function(x) table0(grepl("y", x[, s8bq1])))))
# leave monga out
meal3 <- c(asn(table0(grepl("3", sec3b[[1]][, s23b_1]) & grepl("3", sec3b[[1]][, s23b_2]))
         asn(lapply(sec3b[2:3], function(x) table0(grepl("y", x[, s8bq1])))))
meal3 <- matrix(meal3, byrow = T, ncol = 2)
dimnames(meal3) <- list(paste0("rd", 1:3), c("FALSE", "TRUE"))
meal3
```

```
    FALSE TRUE
rd1  1926  257
rd2  1218  854
rd3   984 1105
```

```
iiD1 <- sec3b[[1]][, receivedCredit]
iiD2 <- sec3b[[2]][, receivedCredit]
iiD3 <- sec3b[[3]][, receivedCredit]
iiI1 <- grepl("treated", sec3b[[1]][, assignment])
iiI2 <- grepl("treated", sec3b[[2]][, assignment])
iiI3 <- grepl("treated", sec3b[[3]][, assignment])
meal3D1 <-
        c(asn(table0(grepl("3", sec3b[[1]][iiD1, s23b_1]) & grepl("3", sec3b[[1]][iiD1, s2
        asn(table0(grepl("y", sec3b[[2]][iiD2, s8bq1]))),
        asn(table0(grepl("y", sec3b[[3]][iiD3, s8bq1]))))
meal3D0 <-
        c(asn(table0(grepl("3", sec3b[[1]][!iiD1, s23b_1]) & grepl("3", sec3b[[1]][!iiD1,
        asn(table0(grepl("y", sec3b[[2]][!iiD2, s8bq1]))),
        asn(table0(grepl("y", sec3b[[3]][!iiD3, s8bq1]))))
meal3I1 <-
        c(asn(table0(grepl("3", sec3b[[1]][iiI1, s23b_1]) & grepl("3", sec3b[[1]][iiI1, s2
        asn(table0(grepl("y", sec3b[[2]][iiI2, s8bq1]))),
        asn(table0(grepl("y", sec3b[[3]][iiI3, s8bq1]))))
meal3I0 <-
        c(asn(table0(grepl("3", sec3b[[1]][!iiI1, s23b_1]) & grepl("3", sec3b[[1]][!iiI1,
        asn(table0(grepl("y", sec3b[[2]][!iiI2, s8bq1]))),
        asn(table0(grepl("y", sec3b[[3]][!iiI3, s8bq1]))))
meal3D1 <- matrix(meal3D1, byrow = T, ncol = 2)
meal3D0 <- matrix(meal3D0, byrow = T, ncol = 2)
meal3I1 <- matrix(meal3I1, byrow = T, ncol = 2)
meal3I0 <- matrix(meal3I0, byrow = T, ncol = 2)
dimnames(meal3D1) <- dimnames(meal3D0) <-
dimnames(meal3I1) <- dimnames(meal3I0) <-
        list(paste0("rd", 1:3), c("FALSE", "TRUE"))
meal3DI <- data.table(rbind(repseq(c("D=1", "D=0", "I=1", "I=0"), 2),
        cbind(meal3D1, meal3D0, meal3I1, meal3I0)))
meal3DI
```

```
   FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE
1:   D=1  D=1   D=0  D=0   I=1  I=1   I=0  I=0
2:  1526  206   400   51   869  118  1057  139
3:  1017  685   201  169   573  394   645  460
4:   793  924   191  181   466  508   518  597
```

```
#z.0 <- parse(text = "s8bq")
#sec3b[[1]][, zval := eval(z.0)]
```

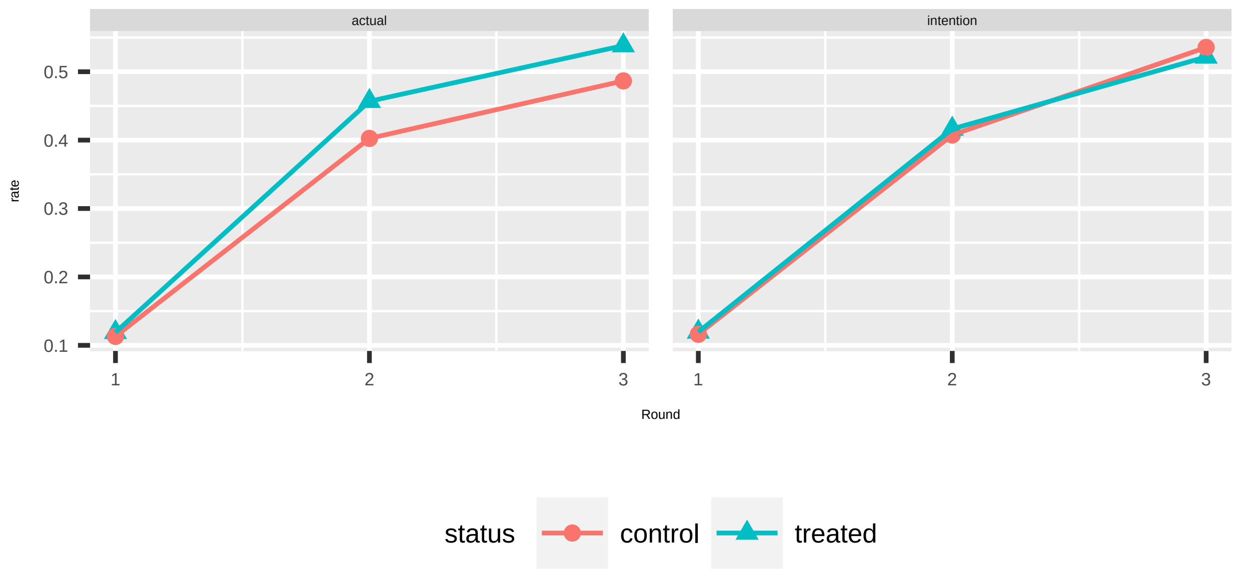

Figure 1   3 meals per day

Given the questions are different, it is not surprising that we have different proportion of subjects with three meals per day. Despite this limitation, we have an increasing food consumption security which is promising.

Form data for regression.

```
s3b ← merge(sec3b[[1]][, .(gid, hhid, s23b_1, s23b_2, arm, assignment,
        disbursed.1, purchased.1, receivedCredit,
        daysFromStart, daysSince2014.1, intDate.1)],
        sec3b[[2]][, .(gid, hhid, s8bq1, arm, assignment,
        disbursed.2, purchased.2, receivedCredit, daysSince2014.2, intDate.2)],
        by = c("gid", "hhid", "arm", "assignment"), all = T, , suffixes = c(".1", ".2"))
s3b ← merge(s3b, sec3b[[3]][, .(gid, hhid, s8bq1, arm, assignment,
        disbursed.3, purchased.3, receivedCredit, daysSince2014.3, intDate.3)],
        by = c("gid", "hhid", "arm", "assignment"), all = T, suffixes = c(".2", ".3"))
setnames(s3b, "receivedCredit", "receivedCredit.3")
dim(s3b); dim(s3b ← s3b[!is.na(hhid) | !is.na(gid), ])
```

```
[1] 2215   24
```

```
[1] 2215   24
```

```
s3b[, c("disbursed.1", "purchased.1", "receivedCredit.1") := F]
if (nrow(s3b[is.na(assignment), ]) > 0)
        s3b[is.na(assignment), assignment := "drop out"]
dim(s3b ← s3b[!duplicated(s3b), ])
```

```
[1] 2215   24
```

3 meals per day in regular times for rd 1. For rd 2, 3, yes to the question.

```
s3b[, c("m3.1", "m3.2", "m3.3") :=
        list(grepl("3", s3b[, s23b_1]) & grepl("3", s3b[, s23b_2]),
        grepl("y", s3b[, s8bq1.2]),
        grepl("y", s3b[, s8bq1.3]))]
```

```
s3b[is.na(s23b_1) | is.na(s23b_2), m3.1 := NA]
s3b[is.na(s3b[, s8bq1.2]), m3.2 := NA]
s3b[is.na(s3b[, s8bq1.3]), m3.3 := NA]
```

Rescale days by 100. Note that assignment has empty observations who either group rejected or lost to flood. They form the reference group for assignment (control, treated).

```
s3b[, daysFromStart := daysFromStart/100]
dim(s3b ← s3b[, !grepl("^s\\d", colnames(s3b)), with = F])
```

```
[1] 2215   23
```

```
#s3bl ← reshape(s3b, direction = "long",
#        idvar = c("gid", "hhid", "assignment", "arm"),
#        varying = grepout("\\.\\d", colnames(s3b)))
dim(s3b.comp ← s3b[!is.na(m3.2) & !is.na(m3.3) &
        !is.na(receivedCredit.2) & !is.na(receivedCredit.3), ])
```

```
[1] 2043   23
```

```
s3bl ← reshape(s3b.comp, direction = "long",
        idvar = c("gid", "hhid", "assignment", "arm"),
        varying = grepout("\\.\\d", colnames(s3b.comp)))
m3data ← s3bl[time > 1, ]
setkey(m3data, hhid, time)
table(table(m3data[, hhid]))
```

```

   2
2043
```

```
m3data[, m3 := m3+0]
m3data[, arm := factor(arm, levels = c("traditional", "large", "large grace", "cow", "lost
m3data[, assigncredit := grepl("tre", assignment) * receivedCredit]
```

m3data: Rd 2-3 data on three meals per day.

```
dm3 ← data.table(m3data[seq(1, nrow(m3data), 2),
        .(gid, arm, assignment, receivedCredit, assigncredit, daysFromStart)],
        m3data[seq(2, nrow(m3data), 2), .(disbursed, purchased, daysSince2014, m3)]-
        m3data[seq(1, nrow(m3data), 2), .(disbursed, purchased, daysSince2014, m3)])
l1 ← glm(m3 ~ arm, data = dm3)
l2 ← glm(m3 ~ assignment, data = dm3)
l3 ← glm(m3 ~ arm*disbursed, data = dm3)
l4 ← glm(m3 ~ assignment + disbursed + assigncredit, data = dm3)
l5 ← glm(m3 ~ arm + daysFromStart, data = dm3)
l6 ← glm(m3 ~ arm*disbursed + daysFromStart, data = dm3)
#p1 ← glm(m3 ~ arm, family=binomial(link="probit"), data = m3data)
linprob ← list(l1, l2, l3, l4, l5, l6)
linest ← lapply(linprob, clx.regobj, Cluster = "gid")
linest ← lapply(linest, function(x) x[, -3])
linest ← tabs2latex(linest)
R2 ← round(asn(lapply(linprob,
        function(x) 1-crossprod(summary(x)$deviance.res)/summary(x)$null.dev)), 3)
en ← asn(lapply(linprob, function(x) length(x$y)))
rn ← rownames(linest)
rn ← gsub("arm|assignment|^se.*", "", rn)
```

Table 1: FD estimates of three meals per day, round 2, 3

rn	(1)	(2)	(3)	(4)	(5)	(6)
(Intercept)	0.118*	−0.060	0.081	−0.060	0.141**	0.051
	(0.062)	(0.111)	(0.066)	(0.111)	(0.055)	(0.055)
large	−0.014		0.067		−0.008	0.124
	(0.099)		(0.096)		(0.097)	(0.086)
large grace	0.093		0.162		0.087	0.197**
	(0.090)		(0.098)		(0.084)	(0.091)
cow	−0.079		−0.051		−0.069	−0.009
	(0.101)		(0.101)		(0.098)	(0.098)
lost to flood	−0.043		−0.005			
	(0.133)		(0.134)			
control		0.211*		0.236**		
		(0.117)		(0.119)		
treated		0.176		0.178		
		(0.117)		(0.141)		
credit			0.147**	−0.062		0.142**
			(0.064)	(0.062)		(0.065)
large * credit			−0.367***			−0.423***
			(0.126)			(0.125)
large grace * credit			−0.230**			−0.265***
			(0.090)			(0.091)
cow * credit			−0.108			−0.122
			(0.134)			(0.136)
treated * credit				−0.006		
				(0.086)		
elapsed days * 100					−0.001	0.006
					(0.006)	(0.009)
R^2	0.01	0.009	0.022	0.011	0.009	0.029
n	2043	2043	1838	1838	1657	1527

Notes: 1. First-difference estimates of having three meals per day using rd 2 and 3 information. Standard errors are clustered at the group level.

2. large, large grace, cow, lost to flood, control, treated, credit are all time invariant and are interacted with a trend term. Regressions (1) - (4) include subjects who group-rejected or lost to flood as a reference group. Regressions (5) - (6) drop subjects who group-rejected or lost to flood and use the subjects who were initially assigned to the control as a reference group.

3. $*$, $**$, $***$ indicate significance levels at 10%, 5%, 1%, respectively.

```
rn ← gsub("assigncredit", "treated * credit", rn)
rn ← gsub("disbursed", "credit", rn)
rn ← gsub(":", " * ", rn)
rn ← gsub("daysFromStart", "elapsed days * 100", rn)
ltab ← rbind(as.matrix(cbind(rn, linest)), c("$R^{2}$", R2),
      c("$n$", en))
write.tablev(latextab(ltab, delimiterline = NULL, alternatecolor2 = "gray90",
      hleft = c("\\footnotesize", rep("\\scriptsize\\hfil$", ncol(ltab)-1)),
      hcenter = c(2.2, rep(1.25, ncol(ltab)-1)),
      hright = c("\\hfill", rep("$", ncol(ltab)-1)),
      adjustlineskip = "-.4ex"),
      paste0(pathsave, "3meals.tex"), colnamestrue = F)
```

We see no impacts of intervention when comparing two perids after the disbursement.

IV Credit use

```
cruse.files ← grepout("21", fn)
cruse.files ← cruse.files[!grepl("com", cruse.files)]
```

File names of rd 3 files are named for the page ordering. For example, ./2/section_21a.prn are the first 2 questions of Section 20, which is named as 21 as it is an unnumbered page that comes right after Section 20. ./2/section_22a.prn is Section 18.

```
setwd(pathsource.mar)
fread(cruse.files[1], integer64 = "double")
```

```
              id s11q1 s11q2
   1:    7010102   Yes   Yes
   2:    7010105   Yes   Yes
   3:    7010106   Yes   Yes
   4:    7010107   Yes   Yes
   5:    7010108   Yes   Yes
  ---
2079: 99081912415   Yes   Yes
2080: 99081912417   Yes   Yes
2081: 99081912418   Yes   Yes
2082: 99081912419   Yes   Yes
2083: 99081912420   Yes   Yes
```

```
fn21 <- grepout("2/section_23.prn|3/section_21", fn)
```

In rd 2, Section 21 is stored under ./2/section_23.prn, in rd 3, ./3/section_21_use_of_credit_1.prn, ./3/section_21_use_of_credit_2.prn.

```
setwd(pathsource.mar)
foldername <- list.dirs(path = ".", recursive = T, full.names = T)
foldername <- foldername[!grepl("add|ori|^\\.$|1$", foldername)]
fn <- unique(list.files(path = foldername, pattern = ".prn$",
        recursive = T, full.names = T))
X = lapply(fn, fread, integer64 = "double")

Cr = copy(X[fn %in% fn21])
Cr <- lapply(Cr, function(x) if (any(grepl("^id$", colnames(x))))
        setnames(x, "id", "hhid") else x)
invisible(lapply(Cr, setkey, hhid))
Cr2 <- Cr[[1]]
setnames(Cr2, colnames(Cr2)[-1],
        paste0("V", 1:(ncol(Cr2)-1), "_", colnames(Cr2)[-1]))
setnames(Cr2, colnames(Cr2),
        gsub("seca3_", "", colnames(Cr2)))
setnames(Cr2, colnames(Cr2),
        gsub("_\\d_|_q\\d_|__", "_", colnames(Cr2)))
setnames(Cr2, colnames(Cr2),
        gsub("_\\d_", "_", colnames(Cr2)))
Cr3 <- Cr[[2]][Cr[[3]]]
setnames(Cr3, colnames(Cr3)[-1],
        paste0("V", 1:(ncol(Cr3)-1), "_", colnames(Cr3)[-1]))
setnames(Cr3, colnames(Cr3),
        gsub("_q.*?_([a-z])", "_\\1", colnames(Cr3)))
setnames(Cr3, colnames(Cr3),
        gsub("_a_", "_", colnames(Cr3)))
Cr3 <- Cr3[!is.na(hhid), ]
setkey(Cr3, hhid, V5_from_when_you_started_1,
        V6_from_when_you_started_2)
#setnames(Cr3, colnames(Cr3)[-1],
#        paste0("V", putzeroontop(1:(ncol(Cr3)-1), totaldigits = 2)))
Cr3[, iga := 1:.N, by = hhid]
Cr3[, igas := .N, by = hhid]
setkey(Cr3, hhid, iga)
```

Merge rd 2 and 3.

```
setnames(Cr2, grepout("from.oth", colnames(Cr2)), "loanFromOther")
setnames(Cr3, grepout("from.oth", colnames(Cr3)), "loanFromOther")
setnames(Cr3, grepout("deta.*j$", colnames(Cr3)), "igaContent")
setnames(Cr3, grepout("deta.*i$", colnames(Cr3)), "specify")
setnames(Cr3, grepout("am.*ed", colnames(Cr3)), "investValue")
setnames(Cr3, grepout("start.*_1", colnames(Cr3)), "startY")
setnames(Cr3, grepout("start.*_2", colnames(Cr3)), "startM")
setnames(Cr3, grepout("du", colnames(Cr3)), "investDuration")
setnames(Cr3, grepout("8.*othe.*g", colnames(Cr3)), "investSame")
setnames(Cr3, grepout("9.how.*", colnames(Cr3)), "investSameNum")
setnames(Cr3, grepout("0.*any.*g", colnames(Cr3)), "investSameExper")
setnames(Cr3, grepout("1.how.*", colnames(Cr3)), "investSameExperNum")
setnames(Cr3, grepout("2.*still", colnames(Cr3)), "investSameStill")
Cr2 ← Cr2[!is.na(hhid), ]
Cr3 ← Cr3[!is.na(hhid), ]
lapply(list(Cr2, Cr3), colnames)
```

```
[[1]]
 [1] "hhid"
 [2] "loanFromOther"
 [3] "V2_way_of_using_guk_credit"
 [4] "V3_use1"
 [5] "V4_use2"
 [6] "V5_use3"
 [7] "V6_other_specifyc"
 [8] "V7_credit_usage1"
 [9] "V8_plan_to_repay"
[10] "V9_plan1"
[11] "V10_plan2"
[12] "V11_plan3"
[13] "V12_other_specifyd"
[14] "V13_years_income_expectation"
[15] "V14_expected_income"
[16] "V15_how_to_spend_extra_income_if_any"
[17] "V16_how_to_spend_extra_income_if_anz"
[18] "V17_ow_to_spend_extra_income_if_anya"
[19] "V18_how_to_spend_specify"
[20] "V19_hh_word_hours_increase"
[21] "V20_other_members_work_hours_increas"
[22] "V21_other_members_work_hours_increat"
[23] "V22_other_members_work_hours_decreas"
[24] "V23_other_members_work_hours_decreat"
[25] "V24_other_members_work_hours_same_mi"
[26] "V25_other_members_work_hours_same_mj"

[[2]]
 [1] "hhid"             "loanFromOther"    "igaContent"
 [4] "specify"          "investValue"      "startY"
 [7] "startM"           "investDuration"   "investSame"
[10] "investSameNum"    "investSameExper"  "investSameExperNum"
[13] "investSameStill"  "iga"              "igas"
```

```
setwd(pathsave)
write.tablev(Cr2, "credit_use_rd_2.prn")
write.tablev(Cr3, "credit_use_rd_2.prn")
```

Intended use of credit, mostly livestock (cows). It is interesting to note that the majority of our subjects choose livestock for an investment.

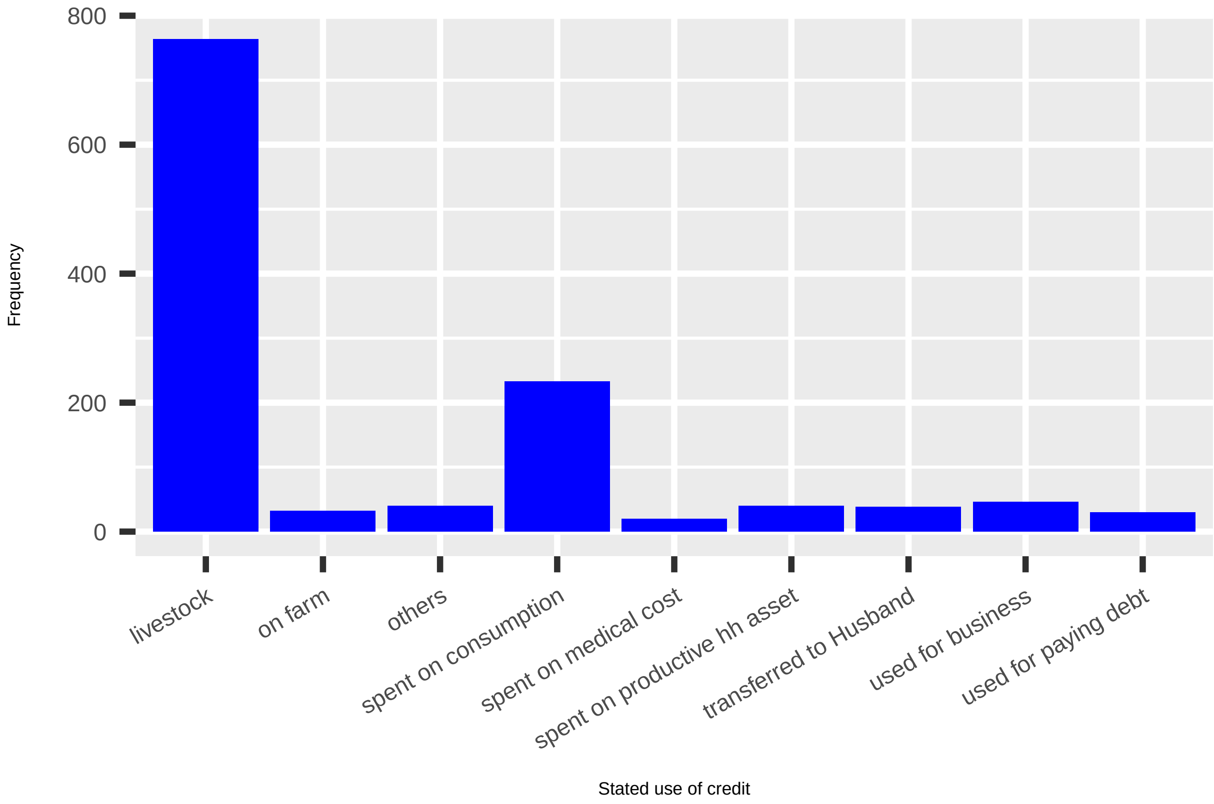

Figure 2 Stated use of credit in rd 2

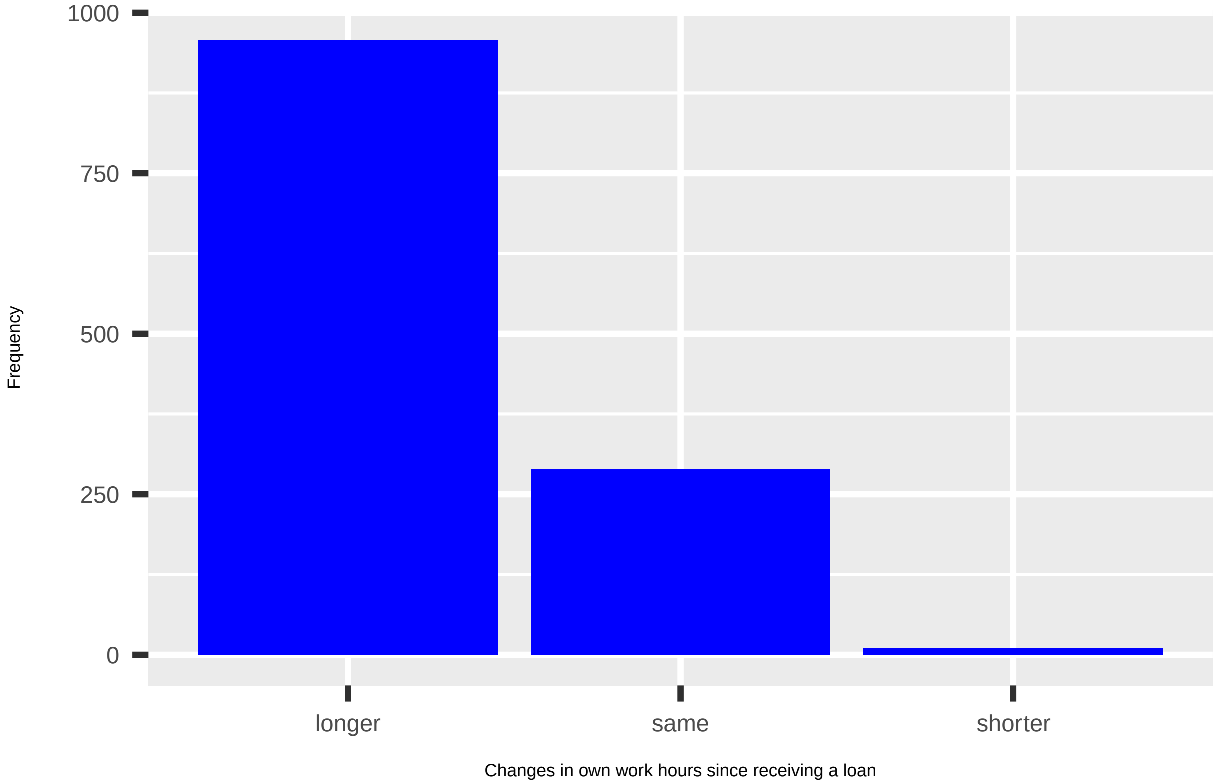

Figure 3 Work hours

Work hours.

V New loans

Loans in rds 1, 2, and 3.

```
(fn20c ← grepout("2/s.*20c|3/s.*19c", fn))
```

```
[1] "./2/section_20c.prn" "./3/section_19c.prn"
```

```
(fn19.1 ← grepout("d/s.*19", fn))
```

```
[1] "./1/combined/s19.prn"
```

```
bo1 = copy(X[[which(fn %in% fn19.1)]])
bo1 ← bo1[!is.na(s19_1_1), ]
bo1 ← bo1[!duplicated(hhid), ]
lendersin1 ← c("other", "relative", "moneylender")
setnames(bo1, paste0("s19_", repseq(1:3, 5), "_", rep(1:5, 3)),
         paste0(c("ask", "askAmount", "cashAmount", "interest", "usage"), ".",
         repseq(lendersin1, 5)))
bo1l ← reshape(bo1, direction = "long", idvar = "hhid",
         varying = grepout("\\.", colnames(bo1)))
setnames(bo1l, "time", "lender")
setkey(bo1l, hhid, lender)
bo1l[, totalSum := sum(cashAmount, na.rm = T), by = hhid]
```

```
Warning in `[.data.table`(bo1l, , `:=`(totalSum, sum(cashAmount, na.rm = T)), : Invalid .:
```

```
bo1l[grepl("oth", lender), lender := "other NGO/MFI"]
bo1l[grepl("rel", lender), lender := "friends, relatives"]
bo1l[grepl("mo", lender), lender := "money lenders"]
Bo1 ← cbind(rd = 1, bo1l)
```

In rd 1, there are only 14 subjects who have borrowed from other NGO/MFI in the last 12 months. Most of the loans are taken from friends, relatives and money lenders, for about 9%, 13% of subjects, respectively.

```
bor1 ← by(bo1l[, cashAmount], bo1l[, lender], destat)
bor1des ← data.frame(rbindlist(lapply(bor1, function(x) data.table(t(matrix(x))))))
dimnames(bor1des) ← list(names(bor1), colnames(bor1[[1]]))
bor1des
```

```
                    min  25\\% median  75\\%   max   mean     std 0s  NAs    n
friends, relatives  100    500   1000  2000 35000 1923.1  2800.3  0 1932 2218
money lenders       100   1500   2000  4000 30000 3344.6  3517.8  0 2041 2218
other NGO/MFI       500   1850   2500  6750 50000 7914.3 13634.9  0 2204 2218
```

```
Bo = copy(X[fn %in% fn20c])
Bo ← lapply(Bo, function(x) if (any(grepl("^id$", colnames(x))))
         setnames(x, "id", "hhid") else x)
invisible(lapply(Bo, setkey, hhid))
Bo ← rbindlist(list(data.table(rd = 2, Bo[[1]]), data.table(rd = 3, Bo[[2]])))
Bo ← Bo[!is.na(hhid), ]
Bo[, bo := 1:.N, by = c("hhid", "rd")]
setkey(Bo, hhid, rd, bo)
Bo[, inkindAmount := (in_kind_amount_4_1) * (in_kind_price_4_1)]
```

```
Bo[is.na(inkindAmount), inkindAmount := 0]
Bo[is.na(cash_tk_4_1), cash_tk_4_1 := 0]
Bo[, cashAmount := cash_tk_4_1]
Bo[, totalSum := sum(cashAmount+inkindAmount), by = c("hhid", "rd")]
Bo[, purpose := pur_loan_4_1]
Bo[grepl("other", purpose), purpose := purpose_of_the_loan_specify_4_1]
Bo[grepl("cow|COW|cuw|cou ?bu|cow ?bu|cwo|gow|cokw|coe|coy|ci=ow", purpose),
        purpose := "buying cows"]
Bo[grepl("goa?t|goad|goot", purpose), purpose := "buying goats"]
Bo[grepl("shee|shepp", purpose), purpose := "buying sheep"]
Bo[grepl("boa?t|boad|ship", purpose), purpose := "buying a boat"]
Bo[grepl("land|lond|lnad", purpose), purpose := "buy/leasing in land"]
Bo[grepl("house", purpose), purpose := "buying a house"]
Bo[grepl("eremny|dowry", purpose), purpose := "ceremony, dowry"]
Bo[grepl("mach", purpose), purpose := "buying machines"]
Bo[grepl("buss?inn?es|trade", purpose), purpose := "business investment"]
table0(Bo[, loan_taken_from_4_1])
```

```
                                                  0
                    799                           1
       Commercial Banks            Government Banks
                      1                           1
            Grameen Bank               Money lender
                     24                         146
  Non-relatives in village Nonrelatives out of village
                     16                           5
      Relatives in village     Relatives out of village
                    540                          15
             Shop owner                      Trader
                    694                          70
           co-operatives         other NGO's(specify)
                      8                        2974
        other(specify)
                     67
```

```
Bo[, lender := tolower(loan_taken_from_4_1)]
Bo[grepl("rela", lender), lender := "friends, relatives"]
Bo[grepl("mo", lender), lender := "money lenders"]
Bo[grepl("0", lender), lender := ""]
Bo[lender == "", lender := NA]
Bo[grepl("sho|tr", lender), lender := "shop owners, traders"]
Bo[grepl("gra|other|co-|ban", lender) | grepl("bra", loan_taken_from_specify_4_1),
        lender := "other NGO/MFI"]
Bo[grepl("GUK|guk|ugk", loan_taken_from_specify_4_1), lender := "GUK"]
table0(Bo[, lender])
```

```
            GUK   friends, relatives        money lenders
            552                  576                  146
  other NGO/MFI shop owners, traders                 <NA>
           2523                  764                  800
```

```
table0(Bo[grepl("other", lender), loan_taken_from_specify_4_1])
```

```
            .     brac  cow  buy
  2520      1        1        1
```

```
table0(Bo[grepl("other", lender), cashAmount])
```

```
     0     300     400     500     600    1000    1500    1600    2000    2500    3000
    10       1       1       3       2       4       5       1       9       1      10
  3500    3800    4000    4500    5000    5600    6000    6500    7000    7500    7800
     6       1      10       2     291       4      14       1     179       1       1
  8000    9000   10000   11300   12000   13000   15000   16000   16800   18000   20000
    39       2      25       1       9       2    1867       1       8       1       5
 22000   70000   80000  150000
     1       1       1       3
```

Append rd 1.

```
setkey(Bo, hhid, rd); setkey(Bo1, hhid, rd)
Bo13 <- rbind(Bo1, Bo, fill = T)
setkey(Bo13, hhid, rd, lender)
```

Merge treatment info.

```
tr01 <- reshape(tr0, direction = "long",
        idvar = c("gid", "hhid", "memstatus", "assignment", "arm"),
        varying = grepout("\\.\\d", colnames(tr0)))
setnames(tr01, "time", "rd")
setkey(Bo13, hhid, rd); setkey(tr01, hhid, rd)
Bot <- tr01[Bo13]
by(Bot[grepl("other|G", lender), cashAmount], Bot[grepl("other|G", lender), rd], destat)
```

```
Bot[grepl("other|G", lender), rd]: 1
     min  25\\% median  75\\%    max    mean      std  0s   NAs     n
v.1.  500   1850    2500   6750  50000  7914.3  13634.9   0  2204  2218
------------------------------------------------------------
Bot[grepl("other|G", lender), rd]: 2
     min  25\\% median  75\\%     max     mean   std  0s  NAs     n
v.1.    0  15000   15000  15000  150000  13085.9  5958   4    0  1285
------------------------------------------------------------
Bot[grepl("other|G", lender), rd]: 3
     min  25\\% median  75\\%     max     mean     std  0s  NAs     n
v.1.    0   7000   15000  15000  150000  12690.9  6282.6   6    0  1790
```

```
Bot[, combined.lender := lender]
Bot[grepl("oth|GU", lender), combined.lender := "NGO/MFI"]
Bot[grepl("shop", lender), combined.lender := "money lenders"]
setwd(pathsave)
write.tablev(Bot, "borrowing_rd_1-3.prn")
```

Plot new loans in each rd. I will combine shop owners/traders with money lenders. I will also combine GUK and other NGO/MFI to NGO/MFI. We also omit zero borrowing from the histogram for clarity.

```
#Bot[, totalSum0 := totalSum]
#Bot[totalSum == 0, totalSum0 := NA]
library(ggplot2)
ggplot(data = subset(Bot, cashAmount > 0), aes(x = cashAmount, fill = combined.lender)) +
        geom_histogram(bins = 20) +
        scale_x_continuous(limits = c(0, 30000)) +
        scale_y_continuous(limits = c(0, 1000)) +
        ylab("Frequency") + xlab("Borrowing (Tk)") + labs(fill = "lenders") +
        facet_wrap(~ rd) +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
```

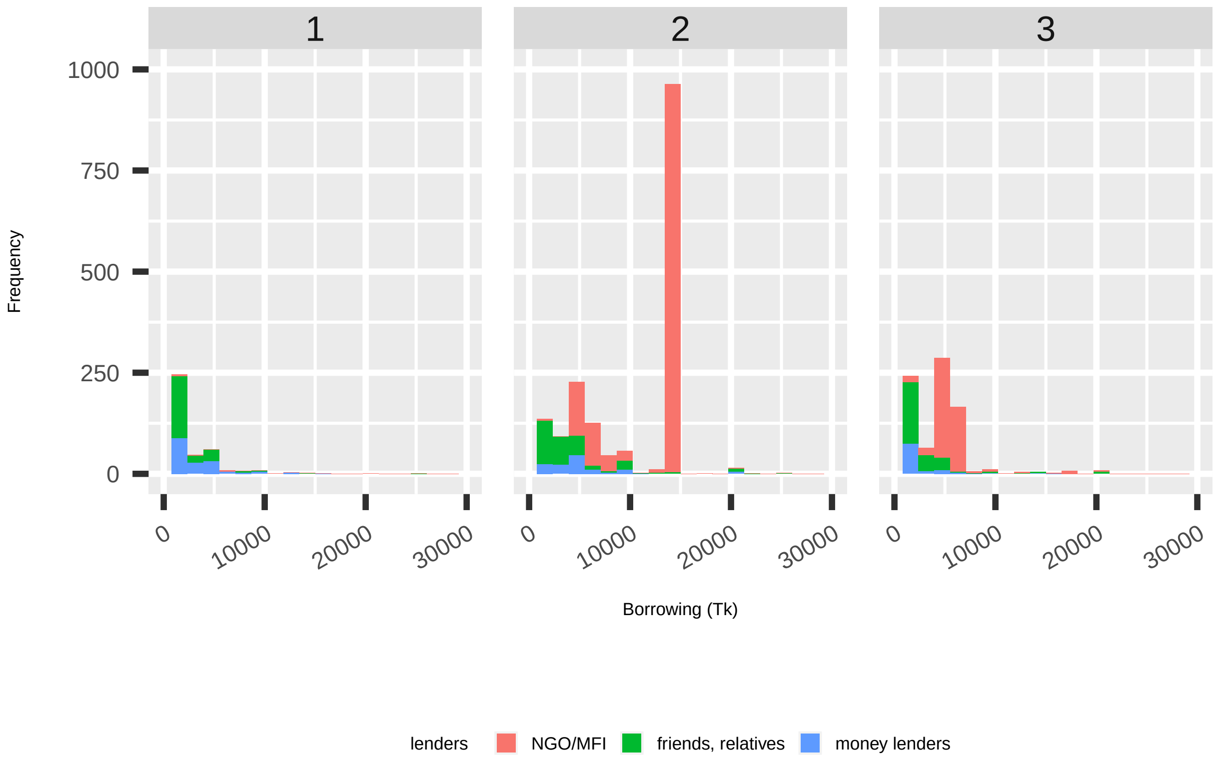

Figure 4 New loans

```
axis.title.x = element_text(size = rel(.25), angle = 0),
axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
axis.text.y = element_text(size = rel(.5), angle = 0),
legend.text = element_text(size=rel(.25)),
legend.position = "bottom",
legend.title = element_text(size = rel(.25)),
legend.key = element_rect(size = rel(.25)),
legend.key.size = unit(.15, "cm"),
strip.text = element_text(size=rel(.5)),
strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

One can see that, in rd 1, there is virtually no borrowing from NGO/MFI among our subjects. This indicates that our study areas are relatively free from other non-indiginous financial intermediaries which allows us to estimate the impacts of our loans without much concerns of treatment contamination. In rd 2, borrowing from NGO/MFI increased rapidly as a result of our intervention. In rds 2 and 3, some individuals report smaller amount, which correspond to our traditional loan arm. It is hard to say that the loans from friends, relatives or money lenders have decreased after our intervetion between rd 1 and rd 2.

VI Assets

Read files.

```
(fn.asset ← grepout("d/s.*14a|d/s.*14b|2/s.*15|3/s.*13", fn))
```

```
[1] "./1/combined/s14a.prn" "./1/combined/s14b.prn" "./2/section_15a.prn"
[4] "./2/section_15b.prn"   "./3/section_13a.prn"   "./3/section_13b.prn"
```

```
setwd(pathsource.mar)
As = copy(X[fn %in% fn.asset])
As ← lapply(As, function(x) if (any(grepl("^id$", colnames(x))))
        setnames(x, "id", "hhid") else x)
As ← lapply(As, function(x) x[!duplicated(x), ])
As ← lapply(As, function(x) x[!is.na(hhid), ])
invisible(lapply(As, setkey, hhid))
invisible(lapply(As[1:2], setkey, hhid, mid))
```

Separate into rds for rd-specfic operations (to be merged back later).

```
As01 ← As[[2]][As[[1]]]
As02 ← As[3:4]
As03 ← As[5:6]
```

Rd 1.

```
As11 ← As01[, grepout("hhid|s14a", colnames(As01)), with = F]
setnames(As11, colnames(As11),
        gsub("s14a_(\\d)_1$", "item.\\1", colnames(As11)))
setnames(As11, colnames(As11),
        gsub("s14a_(\\d)_2$", "own.\\1", colnames(As11)))
setnames(As11, colnames(As11),
        gsub("s14a_(\\d)_3$", "value.\\1", colnames(As11)))
summary(As11[duplicated(As11), ])
```

```
      hhid               item.1              own.1               value.1
 Min.   :7.01e+06   Length:5648        Length:5648        Min.   :1200
 1st Qu.:7.04e+06   Class :character   Class :character   1st Qu.:1500
 Median :7.12e+06   Mode  :character   Mode  :character   Median :1800
 Mean   :1.44e+10                                         Mean   :1700
 3rd Qu.:9.81e+09                                         3rd Qu.:2000
 Max.   :9.91e+10                                         Max.   :2000
                                                          NA's   :5644
    item.2              own.2              value.2          item.3
 Length:5648        Length:5648        Min.   : NA     Length:5648
 Class :character   Class :character   1st Qu.: NA     Class :character
 Mode  :character   Mode  :character   Median : NA     Mode  :character
                                       Mean   :NaN
                                       3rd Qu.: NA
                                       Max.   : NA
                                       NA's   :5648
    own.3              value.3         item.4              own.4
 Length:5648        Min.   : NA     Length:5648        Length:5648
 Class :character   1st Qu.: NA     Class :character   Class :character
 Mode  :character   Median : NA     Mode  :character   Mode  :character
                    Mean   :NaN
                    3rd Qu.: NA
                    Max.   : NA
                    NA's   :5648
    value.4
 Min.   : NA
 1st Qu.: NA
 Median : NA
 Mean   :NaN
 3rd Qu.: NA
 Max.   : NA
 NA's   :5648
```

```
As11[grepl("8207|8220|9416|212016", hhid) & item.1 != "", ]
```

```
          hhid                   item.1  own.1  value.1  item.2  own.2  value.2  item.3
 1: 9808148207                      566      1     1200                      NA
 2: 9808148207                      566      1     1200                      NA
 3: 9808148207 Tube well for drinking    Yes     2000                      NA
 4: 9808148207 Tube well for drinking    Yes     2000                      NA
 5: 9808148220                      566      1     2000                      NA
 6: 9808148220                      566      1     2000                      NA
 7: 9808148220 Tube well for drinking    Yes     1600                      NA
 8: 9808148220 Tube well for drinking    Yes     1600                      NA
 9: 9908169416                      566      1     2000     567      1      200
10: 9908169416                      566      1     2000                      NA
    own.3  value.3  item.4  own.4  value.4
 1:            NA                      NA
 2:            NA                      NA
 3:            NA                      NA
 4:            NA                      NA
 5:            NA                      NA
 6:            NA                      NA
 7:            NA                      NA
 8:            NA                      NA
 9:            NA                      NA
10:            NA                      NA
```

```
As11[grepl(8207, hhid), item.2 := .SD[.N, item.1]]
As11[grepl(8207, hhid), own.2 := .SD[.N, own.1]]
As11[grepl(8207, hhid), value.2 := .SD[.N, value.1]]
As11[grepl(8220, hhid), item.2 := .SD[.N, item.1]]
As11[grepl(8220, hhid), own.2 := .SD[.N, own.1]]
As11[grepl(8220, hhid), value.2 := .SD[.N, value.1]]
As11 ← As11[!duplicated(As11[, hhid]), ]
As12 ← As01[, grepout("hhid|s14b", colnames(As01)), with = F]
setnames(As12, colnames(As12),
        gsub("s14b_(\\d)_1$", "item.\\1", colnames(As12)))
setnames(As12, colnames(As12),
        gsub("s14b_(\\d)_2$", "own.\\1", colnames(As12)))
setnames(As12, colnames(As12),
        gsub("s14b_(\\d)_4$", "value.\\1", colnames(As12)))
setnames(As12, colnames(As12),
        gsub("s14b_(\\d)_3$", "ownership.\\1", colnames(As12)))
setnames(As12, colnames(As12),
        gsub("s14b_(\\d)_5$", "rental.\\1", colnames(As12)))
summary(As12[duplicated(As12), ])
```

```
      hhid              item.1              own.1             ownership.1
 Min.   :7.01e+06   Length:4940        Length:4940        Min.   :  1.0
 1st Qu.:7.04e+06   Class :character   Class :character   1st Qu.: 75.2
 Median :7.13e+06   Mode  :character   Mode  :character   Median :100.0
 Mean   :1.48e+10                                         Mean   : 75.2
 3rd Qu.:9.81e+09                                         3rd Qu.:100.0
 Max.   :9.91e+10                                         Max.   :100.0
                                                          NA's   :4936
    value.1        rental.1        item.2              own.2
 Min.   : 300   Min.   : NA   Length:4940        Length:4940
 1st Qu.: 345   1st Qu.: NA   Class :character   Class :character
 Median : 380   Median : NA   Mode  :character   Mode  :character
 Mean   : 665   Mean   :NaN
 3rd Qu.: 700   3rd Qu.: NA
 Max.   :1600   Max.   : NA
```

```
NA's    :4936   NA's    :4940
 ownership.2       value.2        rental.2        item.3
Min.   :100     Min.   :400     Min.   : NA     Length:4940
1st Qu.:100     1st Qu.:400     1st Qu.: NA     Class :character
Median :100     Median :400     Median : NA     Mode  :character
Mean   :100     Mean   :400     Mean   :NaN
3rd Qu.:100     3rd Qu.:400     3rd Qu.: NA
Max.   :100     Max.   :400     Max.   : NA
NA's   :4939    NA's   :4939    NA's   :4940
   own.3              ownership.3      value.3      rental.3
Length:4940        Min.   : NA     Min.   : NA     Mode:logical
Class :character   1st Qu.: NA     1st Qu.: NA     NA's:4940
Mode  :character   Median : NA     Median : NA
                   Mean   :NaN     Mean   :NaN
                   3rd Qu.: NA     3rd Qu.: NA
                   Max.   : NA     Max.   : NA
                   NA's   :4940    NA's   :4940
   item.4              own.4            ownership.4      value.4
Length:4940        Length:4940        Min.   : NA     Min.   : NA
Class :character   Class :character   1st Qu.: NA     1st Qu.: NA
Mode  :character   Mode  :character   Median : NA     Median : NA
                                      Mean   :NaN     Mean   :NaN
                                      3rd Qu.: NA     3rd Qu.: NA
                                      Max.   : NA     Max.   : NA
                                      NA's   :4940    NA's   :4940
rental.4
Mode:logical
NA's:4940
```

```
As12[grepl(8207, hhid), item.2 := .SD[.N, item.1]]
As12[grepl(8207, hhid), own.2 := .SD[.N, own.1]]
As12[grepl(8207, hhid), ownership.2 := .SD[.N, ownership.1]]
As12[grepl(8207, hhid), value.2 := .SD[.N, value.1]]
As12[grepl(8220, hhid), item.2 := .SD[.N, item.1]]
As12[grepl(8220, hhid), own.2 := .SD[.N, own.1]]
As12[grepl(8220, hhid), ownership.2 := .SD[.N, ownership.1]]
As12[grepl(8220, hhid), value.2 := .SD[.N, value.1]]
As12[grepl(8220, hhid), item.3 := .SD[.N, item.2]]
As12[grepl(8220, hhid), own.3 := .SD[.N, own.2]]
As12[grepl(8220, hhid), ownership.3 := .SD[.N, ownership.2]]
As12[grepl(8220, hhid), value.3 := .SD[.N, value.2]]
As12 ← As12[!duplicated(As12[, hhid]), ]
setnames(As11, colnames(As11), gsub("\\.(\\w)", ".1\\1", colnames(As11)))
setnames(As12, colnames(As12), gsub("\\.(\\w)", ".2\\1", colnames(As12)))
As11 ← reshape(As11, direction = "long", idvar = "hhid",
        varying = grepout("\\.\\d", colnames(As11)))
As12 ← reshape(As12, direction = "long", idvar = "hhid",
        varying = grepout("\\.\\d", colnames(As12)))
As1 ← rbind(As11, As12, fill = T)
As1[, time := NULL]
As1 ← As1[!(is.na(item) | item == ""), ]
As1[, assetNumber := 1:.N, by = hhid]
setkey(As1, hhid, assetNumber)
As1[, totalSum := sum(value, na.rm = T), by = hhid]
```

Rd 2.

```
As21 ← As02[[1]]
setnames(As21, colnames(As21),
```

```
        gsub("sec21_item.*$", "item.1", colnames(As21)))
setnames(As21, colnames(As21),
        gsub("sec21_oth.*$", "specify.1", colnames(As21)))
setnames(As21, colnames(As21),
        gsub("cu.*$", "currentStatus.1", colnames(As21)))
setnames(As21, colnames(As21),
        gsub("dec.*$", "amount.1", colnames(As21)))
setnames(As21, colnames(As21),
        gsub("^ta.*$", "value.1", colnames(As21)))
setnames(As21, colnames(As21),
        gsub("^pu.*$", "lastYear.1", colnames(As21)))
summary(As21[duplicated(As21), ])
```

```
      hhid          item.1             specify.1          currentStatus.1
 Min.   : NA    Length:0           Length:0           Length:0
 1st Qu.: NA    Class :character   Class :character   Class :character
 Median : NA    Mode  :character   Mode  :character   Mode  :character
 Mean   :NaN
 3rd Qu.: NA
 Max.   : NA
    amount.1       value.1       lastYear.1
 Min.   : NA    Min.   : NA    Length:0
 1st Qu.: NA    1st Qu.: NA    Class :character
 Median : NA    Median : NA    Mode  :character
 Mean   :NaN    Mean   :NaN
 3rd Qu.: NA    3rd Qu.: NA
 Max.   : NA    Max.   : NA
```

```
As22 = As02[[2]]
setnames(As22, colnames(As22),
        gsub("sec22_co.*$", "item.2", colnames(As22)))
setnames(As22, colnames(As22),
        gsub("sec22_oth.*$", "specify.2", colnames(As22)))
setnames(As22, colnames(As22),
        gsub("^cu.*$", "currentStatus.2", colnames(As22)))
setnames(As22, colnames(As22),
        gsub("^how.*$", "amount.2", colnames(As22)))
setnames(As22, colnames(As22),
        gsub(".*po.*$", "ownership.2", colnames(As22)))
setnames(As22, colnames(As22),
        gsub(".*taka.*$", "value.2", colnames(As22)))
setnames(As22, colnames(As22),
        gsub(".*rented.*$", "rental.2", colnames(As22)))
summary(As22[duplicated(As22), ])
```

```
      hhid          item.2             specify.2          currentStatus.2
 Min.   : NA    Length:0           Length:0           Length:0
 1st Qu.: NA    Class :character   Class :character   Class :character
 Median : NA    Mode  :character   Mode  :character   Mode  :character
 Mean   :NaN
 3rd Qu.: NA
 Max.   : NA
    amount.2     ownership.2      value.2         rental.2
 Min.   : NA    Min.   : NA    Min.   : NA    Min.   : NA
 1st Qu.: NA    1st Qu.: NA    1st Qu.: NA    1st Qu.: NA
 Median : NA    Median : NA    Median : NA    Median : NA
 Mean   :NaN    Mean   :NaN    Mean   :NaN    Mean   :NaN
 3rd Qu.: NA    3rd Qu.: NA    3rd Qu.: NA    3rd Qu.: NA
```

```
Max.   : NA    Max.   : NA    Max.   : NA    Max.   : NA
```

```
As21 ← reshape(As21, direction = "long", idvar = "hhid",
        varying = grepout("\\.\\d", colnames(As21)))
As22 ← reshape(As22, direction = "long", idvar = "hhid",
        varying = grepout("\\.\\d", colnames(As22)))
As2 ← rbind(As21, As22, fill = T)
As2[, time := NULL]
As2 ← As2[!(is.na(item) | item == ""), ]
As2[, assetNumber := 1:.N, by = hhid]
setkey(As2, hhid, assetNumber)
As2[, totalSum := sum(value, na.rm = T), by = hhid]
```

Rd 3.

```
lapply(As03, colnames)
```

```
[[1]]
[1] "hhid"                         "sec21_item_code"
[3] "sec21_other_specifzz"         "current_status"
[5] "decimal"                      "taka"
[7] "purchased_in_last_one_year"

[[2]]
[1] "hhid"                       "sec22_code"
[3] "sec22_others_specifz"       "current_statut"
[5] "how_many"                   "sec22_portion_owned"
[7] "sec22_value_in_taka"        "sec22_rented_amount_in_tk"
```

```
invisible(lapply(As03, function(x) setnames(x, grepout("code", colnames(x)), "item")))
invisible(lapply(As03, function(x) setnames(x, grepout("spec", colnames(x)), "specify")))
invisible(lapply(As03, function(x) setnames(x, grepout("curr", colnames(x)), "currentStatu
invisible(lapply(As03, function(x) setnames(x, grepout("taka", colnames(x)), "value")))
invisible(lapply(As03, function(x) setnames(x, grepout("deci|many", colnames(x)), "amount"
setnames(As03[[1]], "purchased_in_last_one_year", "lastYear")
setnames(As03[[2]], c("sec22_portion_owned", "sec22_rented_amount_in_tk"),
        c("ownership", "rental"))
As3 ← rbindlist(As[5:6], fill = T)
As3 ← As3[!(is.na(item) | item == ""), ]
As3[, assetNumber := 1:.N, by = hhid]
setkey(As3, hhid, assetNumber)
As3[, totalSum := sum(value, na.rm = T), by = hhid]
```

Bind all 3 rds together.

```
Aslist ← list(cbind(rd = 1, As1), cbind(rd = 2, As2), cbind(rd = 3, As3))
(As ← rbindlist(Aslist, fill = T))
```

```
       rd        hhid                   item own value ownership rental
    1:  1     7010102 Tube well for drinking Yes  1500        NA     NA
    2:  1     7010102              Hand pump Yes  1500       100     NA
    3:  1     7010102   Sickle/Dao/Axe/Spade Yes   300       100     NA
    4:  1     7010103 Tube well for drinking Yes   700        NA     NA
    5:  1     7010103              Hand pump Yes   700       100     NA
   ---
22840:  3 99081912420 tube well for drinking  NA  1600        NA     NA
22841:  3 99081912420           mobile phone  NA  1400        NA     NA
22842:  3 99081912420                 others  NA   400        NA     NA
22843:  3 99081912420            fishing net  NA   250       100     NA
```

```
22844:   3 99081912420    sickle/dao/axe/spade   NA    400        100       NA
       assetNumber  totalSum  specify        currentStatus  amount  lastYear
    1:           1      3300       NA                   NA      NA        NA
    2:           2      3300       NA                   NA      NA        NA
    3:           3      3300       NA                   NA      NA        NA
    4:           1      1900       NA                   NA      NA        NA
    5:           2      1900       NA                   NA      NA        NA
   ---
22840:           2      5950          bought in last year       1       yes
22841:           3      5950          bought in last year       1       yes
22842:           4      5950          bought in last year       1       yes
22843:           5      5950           From previous year       2        NA
22844:           6      5950           From previous year       2        NA
```

```
setwd(pathsave)
write.tablev(As, "asset_holding_rd_1-3.prn")
```

```
Asset ← copy(As[!(rd > 1 & grepl("n", lastYear)), ])
Asset[, assetNumber := 1:.N, by = c("hhid", "rd")]
Asset[, numberOfAssets := .N, by = c("hhid", "rd")]
Asset[, totalSum := sum(value, na.rm = T), by = c("hhid", "rd")]
as0 ← unique(Asset[, .(hhid, rd, totalSum)])
setkey(as0, hhid, rd)
as0[, rd := rd + 1]
as0 ← as0[rd < 4, ]
as1 ← copy(as0)
as1[, rd := rd + 1]
as1 ← as1[rd < 4, ]
setnames(as0, "totalSum", "prevSum.1")
setnames(as1, "totalSum", "prevSum.2")
as0[, prevassetNPV.1 := prevSum.1 * .95]
as1[, prevassetNPV.2 := prevSum.2 * .95^(2)]
setkey(as0, hhid, rd); setkey(as1, hhid, rd)
as01 ← as1[as0]
as01[is.na(prevassetNPV.2), prevassetNPV.2 := 0]
as01[, prevassetNPV := prevassetNPV.1 + prevassetNPV.2]
setkey(as01, hhid, rd); setkey(Asset, hhid, rd)
Asset01 ← merge(Asset, as01, by = c("hhid", "rd"), all = T)
Asset01[is.na(prevassetNPV), prevassetNPV := 0]
Asset01[, assetNPV := totalSum + prevassetNPV]
# merge with treatment info
setkey(Asset01, hhid, rd); setkey(tr01, hhid, rd)
Asset01t ← tr01[Asset01]
```

Drop rd 2 and 3 assets that were not bought in the lastYear to avoid double counting.

```
asset ← Asset01t[assetNumber == 1, ]
asset.ss ← subset(asset, assetNPV > 0 & !is.na(receivedCredit))
asset.cross ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, receivedCredit=asset.ss$receivedCredit), median)
asset.cross2 ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, receivedCredit=asset.ss$receivedCredit), mean)
vline.dat ← data.frame(rd = rep(1:3, 2), receivedCredit = repseq(c(F, T), 3))
vline.dat ← cbind(vline.dat, median = c(asset.cross), mean = c(asset.cross2))
library(ggplot2)
ggplot(data = subset(asset, assetNPV > 0 & !is.na(receivedCredit)),
        aes(x = assetNPV, fill = arm)) +
        geom_histogram(bins = 20) +
```

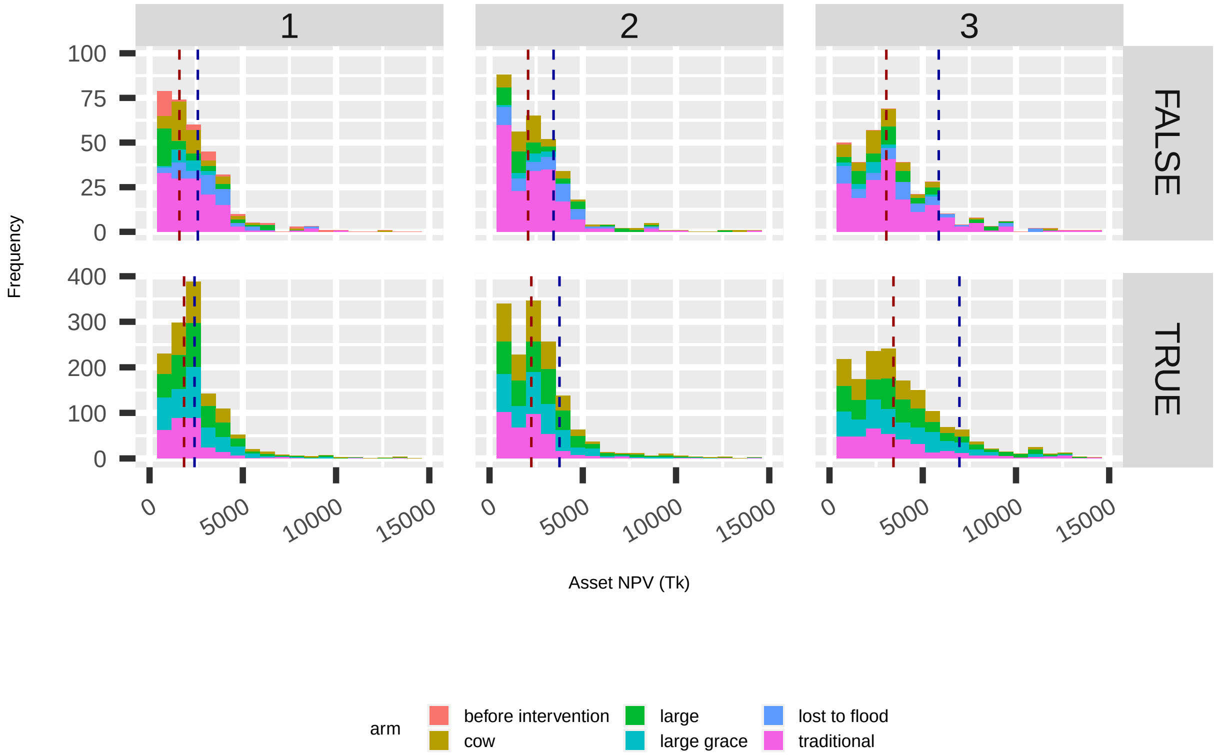

Figure 5 Assets by eventual treatmemt status

```
        scale_x_continuous(limits = c(0, 15000)) +
        #scale_y_continuous(limits = c(0, 1000)) +
        geom_vline(aes(xintercept = median),
                colour="#990000", linetype="dashed", size = .2, data=vline.dat) +
        geom_vline(aes(xintercept = mean),
                colour="#000099", linetype="dashed", size = .2, data=vline.dat) +
        ylab("Frequency") + xlab("Asset NPV (Tk)") + labs(fill = "arm") +
        facet_grid(receivedCredit ~ rd, scales = "free_y") +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

The histogram is created by imputing the NPV of household assets by assuming an annual 5% depreciation rate. We see that, at rd 1, there is no difference in mean of asset holding, while the medians are different. Interestingly, the median difference is preserved in the later rounds. In the meantime, means are not different in rd 1 yet they come to differ in later rounds. The subjects who actually received credits have higher mean asset holding. Given that the median diffrences are unchanged, this indicates that the upper half of the treated asset holders are getting better than the control.

To align dates of receiving credits for the subjects who did not, we use the median daysFromStart.

```
setkey(Asset01t, gid, hhid)
# surround with as.double becuse median function returns various types ....
# see SO: https://stackoverflow.com/questions/12125364/why-does-median-trip-up-data-table-
Asset01t[, medianElapsedDaysOfGroup :=
        as.double(median(elapsed, na.rm = T)), by = gid]
Asset01t[, meanElapsedDaysOfGroup := mean(elapsed, na.rm = T), by = gid]
Asset01t[, elaspsedGroupMedian := "early"]
Asset01t[medianElapsedDaysOfGroup -
        median(medianElapsedDaysOfGroup, na.rm = T) ≤ 0,
        elaspsedGroupMedian := "late"]
Asset01t[, elaspsedGroupMean := "early"]
Asset01t[meanElapsedDaysOfGroup -
        mean(meanElapsedDaysOfGroup, na.rm = T) ≤ 0,
        elaspsedGroupMean := "late"]
Asset01t[, elaspsedGroupMedian := factor(elaspsedGroupMedian)]
Asset01t[, elaspsedGroupMean := factor(elaspsedGroupMean)]
```

```
asset ← Asset01t[assetNumber == 1, ]
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(disbursed))
asset.cross ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, disbursed=asset.ss$disbursed), median)
asset.cross2 ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, disbursed=asset.ss$disbursed), mean)
vline.dat ← data.frame(rd = rep(1:3, 2), disbursed = repseq(c(F, T), 3))
vline.dat ← cbind(vline.dat, median = c(asset.cross), mean = c(asset.cross2))
library(ggplot2)
ggplot(data = asset.ss,
        aes(x = assetNPV, fill = arm)) +
        geom_histogram(bins = 20) +
        scale_x_continuous(limits = c(0, 15000)) +
        geom_vline(aes(xintercept = median),
                colour="#990000", linetype="dashed", size = .2, data=vline.dat) +
        geom_vline(aes(xintercept = mean),
                colour="#000099", linetype="dashed", size = .2, data=vline.dat) +
        ylab("Frequency") + xlab("Asset NPV (Tk)") + labs(fill = "arm") +
        facet_grid(disbursed ~ rd, scales = "free_y") +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

In this figure, we dropped observations without gid and disbursed. When intDate is NA (not interviewed), we cannot define disbursement for that round. We know disbursement took place before rd 3, so all assignment = treated have disbursed = T in rd 3.

```
asset ← Asset01t[assetNumber == 1, ]
```

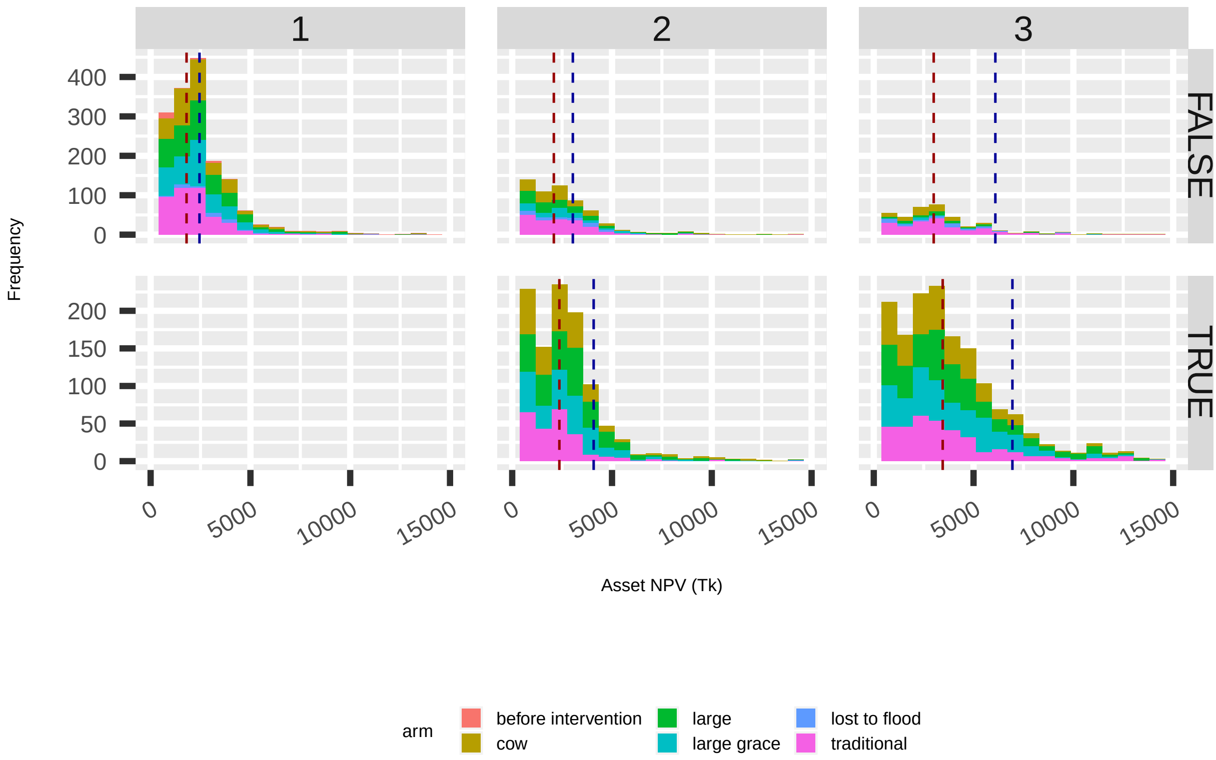

Figure 6   Assets by disbursement status

```
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed) & receivedCredit)
asset.cross ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, elaspsedGroupMedian = asset.ss$elaspsedGroupMedian), median
asset.cross2 ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, elaspsedGroupMedian = asset.ss$elaspsedGroupMedian), mean)
vline.dat ← data.frame(rd = rep(1:3, 2), elaspsedGroupMedian = repseq(c("early", "late"),
vline.dat ← cbind(vline.dat, median = c(asset.cross), mean = c(asset.cross2))
library(ggplot2)
ggplot(data = asset.ss,
        aes(x = assetNPV, fill = arm)) +
        geom_histogram(bins = 20) +
        scale_x_continuous(limits = c(0, 15000)) +
        geom_vline(aes(xintercept = median),
                colour="#990000", linetype="dashed", size = .2, data=vline.dat) +
        geom_vline(aes(xintercept = mean),
                colour="#000099", linetype="dashed", size = .2, data=vline.dat) +
        ylab("Frequency") + xlab("Asset NPV (Tk)") + labs(fill = "arm") +
        facet_grid(elaspsedGroupMedian ~ rd, scales = "free_y") +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
```

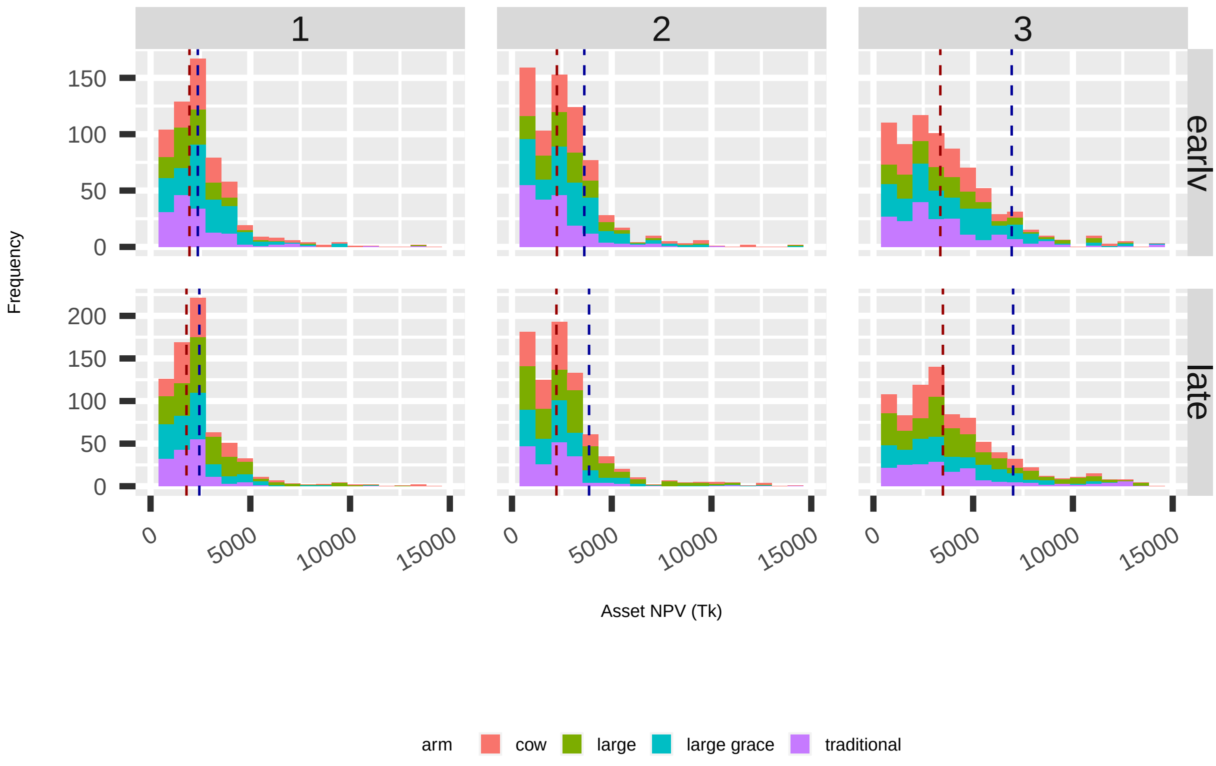

Figure 7 Assets by elapsed days from disbursement, ATT

```
                    strip.text = element_text(size=rel(.5)),
                    strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                    strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))

asset ← Asset01t[assetNumber == 1, ]
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed) & receivedCredit)
library(ggplot2)
ggplot(data = asset.ss, aes(x = elapsed, y = assetNPV)) +
        #geom_jitter(aes(colour = arm, shape = arm), size = .05, width = .1) +
        geom_point(aes(colour = arm, shape = arm), size = .05) +
        scale_shape(solid = F) +
        #scale_y_continuous(limits = c(0, 25000)) +
        scale_y_log10() +
        xlab("elapsed day grouping") + ylab("Asset NPV (Tk)") + labs(fill = "arm") +
        facet_grid(arm ~ rd) +
#       stat_smooth(method = "loess", size = .2, n = 150) +
        geom_smooth(method = "loess", size = .2) +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                    axis.title.x = element_text(size = rel(.25), angle = 0),
                    axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                    axis.text.y = element_text(size = rel(.5), angle = 0),
                    legend.text = element_text(size=rel(.25)),
                    legend.position = "bottom",
                    legend.title = element_text(size = rel(.25)),
                    legend.key = element_rect(size = rel(.25)),
                    legend.key.size = unit(.15, "cm"),
                    strip.text = element_text(size=rel(.5)),
                    strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
```

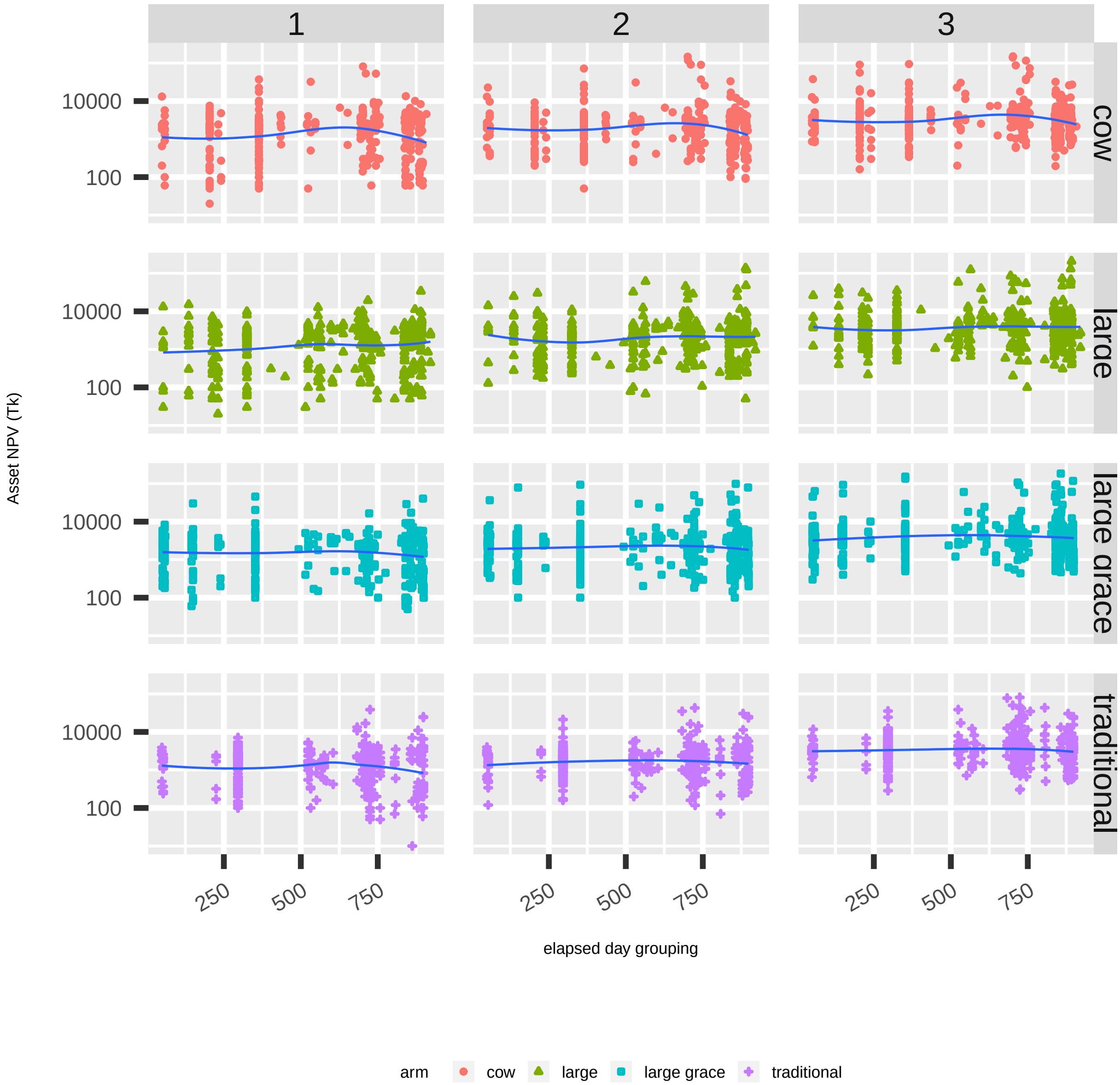

Figure 8 Assets by elapsed days from disbursement, ATT scatter plot

```
strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

In this scatter and loess plots, we put asset values against the treatment exposure, facetted by treatment arms. This aims to mimic ATT under a continuous treatment. The treatment exposure is defined by the elapsed days since receiving a credit. Since the treatment exposure is randomised, this is a statisically valid procedure to observe the treatment response without major comfounding.

This plotting exercise leads one to consider the statisical model underlying the graphs. For an individual i's outcome y_i, the treatment assignment $D_i = 0, 1$ may have an impact on the outcome. The standard Rubin causal model deals with a binary indicator variable for D_i. In our design, we vary the dates of intervention among the subjects. So what we randomly vary is the duration under treatment, or dose exposure, denoted with $D_i(t)$ where t is the calendar date of intervention. On average, there is about 1 year difference in t within a cluster of 20 subjects. Given that we randomise the calendar dates of starting the intevention, we can assume actual duration $t \in [t_0, t_1]$ is orthogonal to potential treatment response $y(t)$ for all t. Under the simplest setting, we follow Imbens (2000); Hirano and Imbens (2005); Imai and van Dyk (2004); Egger and von Ehrlich (2013) assume the following conditional orthogonality in the continuous case. Denoting T as a random variable with its realisation written as t, we assume:

$$y(t) \perp T|\mathbf{x}.$$

Hirano and Imbens (2005) shows that this is equivalent to

$$\mathbf{x} \perp 1\{T = t\}|g(t, \mathbf{x})$$

where $g(t, \mathbf{x})$ is a generalised propesity score that gives the density of treatment at t given $\mathbf{x}$. This shows that one can estimate continuous treatment effect by first, estimating GPS g, second, estimate the conditional expectation of outcome as a function of g and $\mathbf{x}$:

$$\beta(t, g) = \mathcal{E}[y|T = t, G = g(t, \mathbf{x})],$$

and then average over g for a given t to obtain the dose-response function

$$\beta(t) = \mathcal{E}[\beta(t, g)|\mathbf{x}].$$

The approach is preceded by applied works related job training duration (Kluve et al., 2012).

We compare the effects of treatment exposure differences within the same group.

```
asset ← Asset01t[assetNumber == 1, ]
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed))
setkey(asset.ss, rd, gid, assignment)
asset.ss[, avgElapsed := mean(elapsed, na.rm = T), by = c("rd", "gid", "assignment")]
asset.ss[, avgElapsed0 := avgElapsed[1], by = c("rd", "gid")]
asset.ss[, avgElapsed1 := avgElapsed[.N], by = c("rd", "gid")]
asset.ss[gid == 70650 & grepl("co", assignment), ]
```

```
          hhid disburseDate purchaseDate   memstatus receivedCredit exist
 1:    7065010   2014-12-09   2014-12-11         old           TRUE   123
 2: 9807065006   2014-12-09   2014-12-11 replacement           TRUE   123
 3: 9807065013   2014-12-09   2014-12-11 replacement           TRUE   123
 4: 9807065019   2014-12-09   2014-12-11 replacement           TRUE   123
 5:    7065010   2014-12-09   2014-12-11         old           TRUE   123
 6: 9807065006   2014-12-09   2014-12-11 replacement           TRUE   123
 7: 9807065013   2014-12-09   2014-12-11 replacement           TRUE   123
 8: 9807065019   2014-12-09   2014-12-11 replacement           TRUE   123
 9:    7065010   2014-12-09   2014-12-11         old           TRUE   123
10: 9807065006   2014-12-09   2014-12-11 replacement           TRUE   123
11: 9807065013   2014-12-09   2014-12-11 replacement           TRUE   123
12: 9807065019   2014-12-09   2014-12-11 replacement           TRUE   123
    inData   gid i.memstatus assignment         arm i.receivedCredit elapsed
 1:   TRUE 70650         old    control traditional             TRUE     296
 2:   TRUE 70650 replacement    control traditional             TRUE     296
 3:   TRUE 70650 replacement    control traditional             TRUE     296
 4:   TRUE 70650 replacement    control traditional             TRUE     296
 5:   TRUE 70650         old    control traditional             TRUE     296
 6:   TRUE 70650 replacement    control traditional             TRUE     296
 7:   TRUE 70650 replacement    control traditional             TRUE     296
 8:   TRUE 70650 replacement    control traditional             TRUE     296
 9:   TRUE 70650         old    control traditional             TRUE     296
10:   TRUE 70650 replacement    control traditional             TRUE     296
11:   TRUE 70650 replacement    control traditional             TRUE     296
12:   TRUE 70650 replacement    control traditional             TRUE     296
    daysFromStart rd disbursed purchased    intDate daysSince2014
 1:           625  1     FALSE     FALSE 2012-11-08          -419
 2:           625  1     FALSE     FALSE       <NA>            NA
 3:           625  1     FALSE     FALSE       <NA>            NA
```

```
 4:           625  1      FALSE      FALSE       <NA>           NA
 5:           625  2      FALSE      FALSE 2004-01-21        -3633
 6:           625  2      FALSE      FALSE 2004-01-11        -3643
 7:           625  2      FALSE      FALSE 2004-01-11        -3643
 8:           625  2      FALSE      FALSE 2004-01-11        -3643
 9:           625  3       TRUE       TRUE 2015-12-17          715
10:           625  3       TRUE       TRUE 2015-12-10          708
11:           625  3       TRUE       TRUE 2015-11-28          696
12:           625  3       TRUE       TRUE 2015-12-12          710
                  item own value ownership rental assetNumber totalSum
 1: Tube well for drinking Yes   700        NA     NA           1     1620
 2:                  566   1  1200        NA     NA           1     1200
 3:                  566   1  2000        NA     NA           1     2300
 4:                  566   1  1600        NA     NA           1     1900
 5:          fishing net  NA    50       100     NA           1      350
 6:   sickle/dao/axe/spade  NA   200       100     NA           1      200
 7:               weeder  NA   100       100     NA           1      400
 8:   sickle/dao/axe/spade  NA   150       100     NA           1      150
 9:   sickle/dao/axe/spade  NA   200       100     NA           1      200
10:   sickle/dao/axe/spade  NA   300       100     NA           1      300
11: tube well for drinking  NA  1200        NA     NA           1     1600
12:   sickle/dao/axe/spade  NA   300       100     NA           1      300
    specify      currentStatus amount lastYear numberOfAssets prevSum.2
 1:      NA                 NA     NA       NA              4        NA
 2:      NA                 NA     NA       NA              1        NA
 3:      NA                 NA     NA       NA              2        NA
 4:      NA                 NA     NA       NA              2        NA
 5:          From previous year      1       NA              2        NA
 6:          From previous year      2       NA              1        NA
 7:          From previous year      1       NA              2        NA
 8:          From previous year      1       NA              1        NA
 9:          From previous year      2       NA              1      1620
10:          From previous year      3       NA              1      1200
11:         bought in last year      1      yes              3      2300
12:          From previous year      3       NA              1      1900
    prevassetNPV.2 prevSum.1 prevassetNPV.1 prevassetNPV assetNPV
 1:             NA        NA             NA         0.00  1620.00
 2:             NA        NA             NA         0.00  1200.00
 3:             NA        NA             NA         0.00  2300.00
 4:             NA        NA             NA         0.00  1900.00
 5:           0.00      1620         1539.0      1539.00  1889.00
 6:           0.00      1200         1140.0      1140.00  1340.00
 7:           0.00      2300         2185.0      2185.00  2585.00
 8:           0.00      1900         1805.0      1805.00  1955.00
 9:        1462.05       350          332.5      1794.55  1994.55
10:        1083.00       200          190.0      1273.00  1573.00
11:        2075.75       400          380.0      2455.75  4055.75
12:        1714.75       150          142.5      1857.25  2157.25
    medianElapsedDaysOfGroup meanElapsedDaysOfGroup elaspsedGroupMedian
 1:                      737                693.795               early
 2:                      737                693.795               early
 3:                      737                693.795               early
 4:                      737                693.795               early
 5:                      737                693.795               early
 6:                      737                693.795               early
 7:                      737                693.795               early
 8:                      737                693.795               early
 9:                      737                693.795               early
10:                      737                693.795               early
11:                      737                693.795               early
12:                      737                693.795               early
```

```
   elaspsedGroupMean avgElapsed avgElapsed0 avgElapsed1
 1:             early        296         296       828.9
 2:             early        296         296       828.9
 3:             early        296         296       828.9
 4:             early        296         296       828.9
 5:             early        296         296       828.9
 6:             early        296         296       828.9
 7:             early        296         296       828.9
 8:             early        296         296       828.9
 9:             early        296         296       828.9
10:             early        296         296       828.9
11:             early        296         296       828.9
12:             early        296         296       828.9
```

```
asset.ss[gid == 70204& rd == 1, .(rd, gid, assignment, avgElapsed, avgElapsed0, avgElapsed
```

```
   rd   gid assignment avgElapsed avgElapsed0 avgElapsed1
 1:  1 70204    control    517.222     517.222       837.1
 2:  1 70204    control    517.222     517.222       837.1
 3:  1 70204    control    517.222     517.222       837.1
 4:  1 70204    control    517.222     517.222       837.1
 5:  1 70204    control    517.222     517.222       837.1
 6:  1 70204    control    517.222     517.222       837.1
 7:  1 70204    control    517.222     517.222       837.1
 8:  1 70204    control    517.222     517.222       837.1
 9:  1 70204    control    517.222     517.222       837.1
10:  1 70204    treated    837.100     517.222       837.1
11:  1 70204    treated    837.100     517.222       837.1
12:  1 70204    treated    837.100     517.222       837.1
13:  1 70204    treated    837.100     517.222       837.1
14:  1 70204    treated    837.100     517.222       837.1
15:  1 70204    treated    837.100     517.222       837.1
16:  1 70204    treated    837.100     517.222       837.1
17:  1 70204    treated    837.100     517.222       837.1
18:  1 70204    treated    837.100     517.222       837.1
19:  1 70204    treated    837.100     517.222       837.1
```

```
asset.ss[, avgNPV := mean(assetNPV/1000, na.rm = T),
         by = c("rd", "gid", "assignment")]
asset.ss[, avgNPV0 := avgNPV[1], by = c("rd", "gid")]
asset.ss[, avgNPV1 := avgNPV[.N], by = c("rd", "gid")]
asset.ss[, avgDiffElapsed := avgElapsed1 - avgElapsed0]
asset.ss[, avgDiffNPV := avgNPV1 - avgNPV0]
setkey(asset.ss, rd, gid, assignment)
dim(asset.sss <- asset.ss[!duplicated(asset.ss[, .(rd, gid, assignment)]), ])
```

```
[1] 543  49
```

```
library(ggplot2)
ggplot(data = asset.sss, aes(x = avgDiffElapsed, y = avgDiffNPV)) +
         geom_point(aes(colour = arm, shape = arm), size = .05) +
         scale_shape(solid = F) +
         xlab("difference in elapsed days") + ylab("difference in mean asset NPV (Tk '000)"
         labs(fill = "arm") + facet_grid( ~ rd) +
#        stat_smooth(method = "loess", size = .2, n = 150) +
         geom_smooth(method = "loess", size = .2) +
         geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed", size = .2) +
         theme(axis.title.y = element_text(size = rel(.25), angle = 90),
```

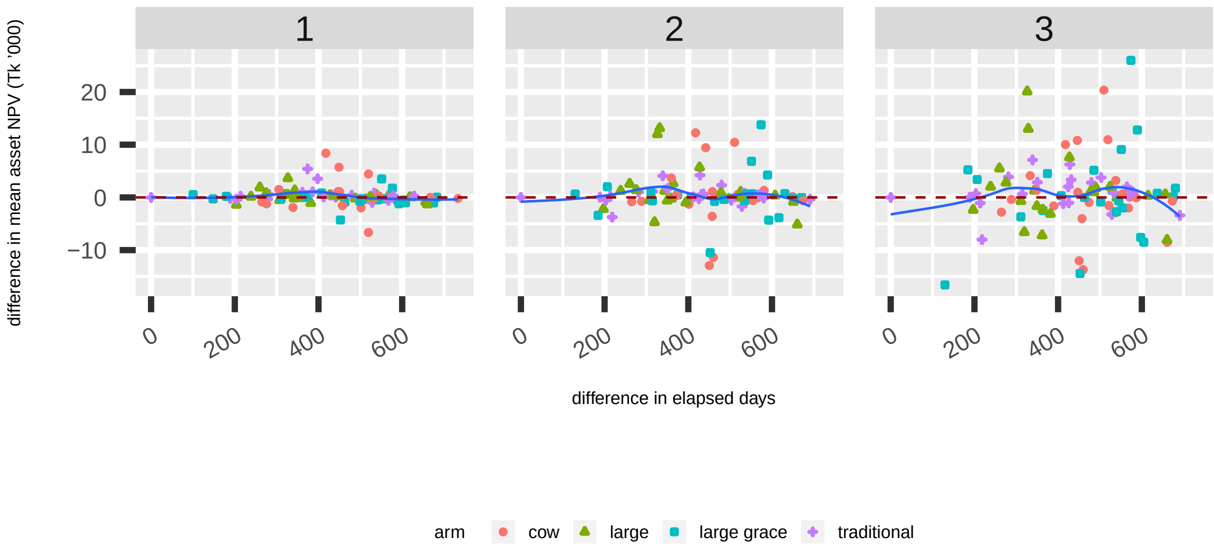

Figure 9 Assets by elapsed days from disbursement within a group

```
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

```
library(ggplot2)
ggplot(data = asset.sss, aes(x = avgDiffElapsed, y = avgDiffNPV)) +
        geom_point(aes(colour = arm, shape = arm), size = .05) +
        scale_shape(solid = F) +
        xlab("difference in elapsed days") + ylab("difference in mean asset NPV (Tk '000)'
        labs(fill = "arm") + facet_grid(arm ~ rd, scale = "free_y") +
#       stat_smooth(method = "loess", size = .2, n = 150) +
        geom_smooth(method = "loess", size = .2) +
        geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed", size = .2) +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

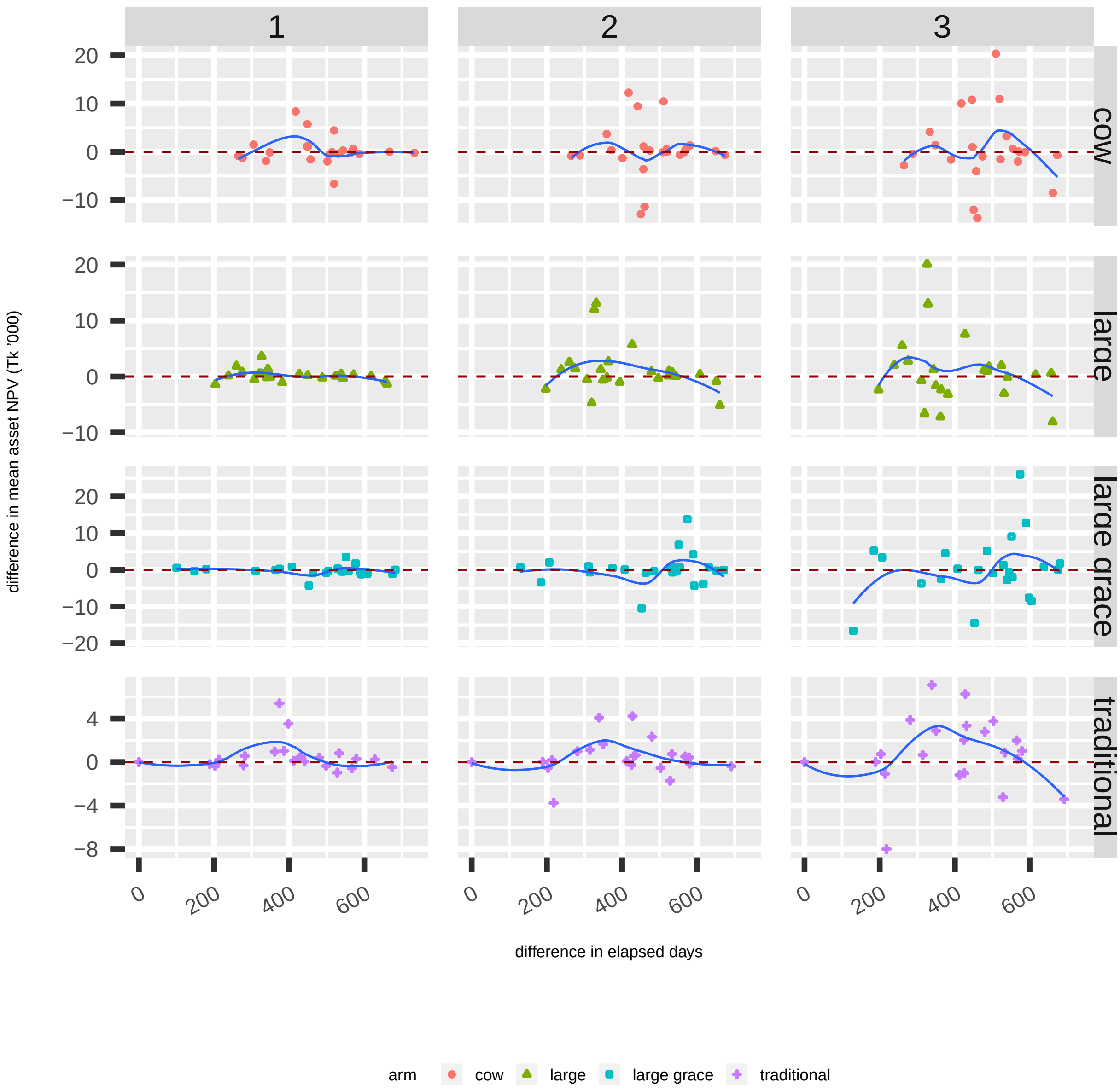

Figure 10 Assets by elapsed days from disbursement, within a group

VII livestock

```
(fn.lvstk ← grepout("d/s08|2/s.*9_|3/s.*_8", fn))
```

```
[1] "./1/combined/s08a.prn" "./1/combined/s08b.prn" "./2/section_9_1.prn"
[4] "./2/section_9_2.prn"   "./2/section_9_3.prn"   "./3/section_8.prn"
[7] "./3/section_8a.prn"    "./3/section_8b.prn"
```

```
setwd(pathsource.mar)
Ls = copy(X[fn %in% fn.lvstk])
Ls ← lapply(Ls, function(x) if (any(grepl("^id$", colnames(x))))
        setnames(x, "id", "hhid") else x)
Ls ← lapply(Ls, function(x)
```

```
        x[!apply(is.na(x[, -grep("hh|mid|u_id", colnames(x)), with = F]) |
                x[, -grep("hh|mid|u_id", colnames(x)), with = F] == "" |
                x[, -grep("hh|mid|u_id", colnames(x)), with = F] == "No", 1, all), ])
Ls <- lapply(Ls, a2b, a = NA, b = 0)
Ls <- lapply(Ls, a2b, a = "", b = 0)
Ls[1:2] <- lapply(Ls[1:2], setkey, hhid, mid)
Ls[-(1:2)] <- lapply(Ls[-(1:2)], setkey, hhid)
Ls1 <- merge(Ls[[1]], Ls[[2]], by = c("hhid", "mid"), all = T)
Ls2 <- merge(Ls[[3]], Ls[[4]], by = "hhid", all = T)
Ls2 <- merge(Ls2, Ls[[5]], by = "hhid", all = T)
Ls2 <- Ls2[!duplicated(Ls2[, .(hhid, s17a_code)]), ]
Ls3 <- merge(Ls[[6]], Ls[[7]], by = "hhid", all = T)
Ls3 <- merge(Ls3, Ls[[8]], by = "hhid", all = T)
Ls3 <- Ls3[!duplicated(Ls3[, .(hhid, s17a_code)]), ]
```

Rd 1.

```
# M: managing, L: leased in
Ls1[, ushiM := s8a_a_2 + s8a_a_3]
Ls1[, calfM := s8a_a_4]
Ls1[, yagiM := s8a_a_5 + s8a_a_6]
Ls1[, ushiL := s8a_b_8 + s8a_b_9]
Ls1[, calfL := s8a_b_10]
Ls1[, yagiL := s8a_b_11]
Ls1 <- a2b(Ls1, NA, 0)
destat(Ls1[, .(ushiM, calfM, yagiM, ushiL, calfL)])
```

```
      min 25\\% median 75\\% max mean std   0s NAs    n
ushiM   0     0      0     1   6  0.4 0.7 1200   0 1780
calfM   0     0      0     0   4  0.3 0.6 1353   0 1780
yagiM   0     0      0     0   8  0.4 1.0 1395   0 1780
ushiL   0     0      0     0   3  0.1 0.4 1630   0 1780
calfL   0     0      0     0   2  0.1 0.3 1680   0 1780
```

```
cpr <- destat(Ls1[s8a_b_15 > 0, s8a_b_15])
cpr2 <- rbind(c(destat(Ls1[s8a_b_15 > 0, s8a_b_15])),
c(destat(Ls1[s8a_b_16 > 0, s8a_b_16])),
c(destat(Ls1[s8a_b_17 > 0, s8a_b_17])))
dimnames(cpr2) <- list(c("female calf", "male calf", "ox"),
        colnames(cpr))
```

Price: female calf, male calf, ox.

```
cpr2
```

```
              min 25\\% median 75\\%   max    mean    std 0s NAs  n
female calf  5000  8000  10000 13500 31000 11244.4 4458.9  0   0 45
male calf    2000 10000  10000 12000 16000 10300.0 3221.0  0   0 25
ox            500  8000  10000 12000 30000 10549.3 5163.6  0   0 73
```

Let the price to be used as median price, and cow price is 15000. Lease share is 50%.

```
destat(Ls1[s8a_b_18 > 0, s8a_b_18])
```

```
     min 25\\% median 75\\% max mean std 0s NAs  n
v.1.  15    50     50    50  60 49.6 4.3  0   0 71
```

```
destat(Ls1[s8a_b_24 > 0, s8a_b_24])
```

```
     min 25\\% median 75\\%  max mean std 0s NAs n
v.1. 9000  9000   9000  9000 9000 9000   0  0   0 2
```

```
destat(Ls2[grepl("a", s17a_code) & s17a_4 > 0, s17a_4])
```

```
     min 25\\% median 75\\%   max    mean    std 0s NAs    n
v.1.   2 15000  17000 20000 60000 17169.1 4582.7  0   0 1266
```

```
destat(Ls3[grepl("cow", s17a_code) & s17a_4 > 0, s17a_4])
```

```
     min 25\\% median 75\\%   max    mean    std 0s NAs    n
v.1.   1 18000  20000 23000 50000 20240.5 5049.1  0   0 1705
```

```
Ls1[, cowValue := ushiM * 15000]
Ls1[, calfValue := calfM * 10000]
Ls1[, cowLValue := ushiL * 15000 * .5]
Ls1[, calfLValue := calfL * 10000 * .5]
```

Goats: Take prices from late rounds. 1900.

```
destat(Ls2[grepl("c", s17a_code) & s17a_4 > 0, s17a_4])
```

```
     min 25\\% median 75\\%   max   mean    std 0s NAs   n
v.1.   2  1500   1900  2800 18000 2161.8 1445.1  0   0 600
```

```
destat(Ls3[grepl("goa", s17a_code) & s17a_4 > 0, s17a_4])
```

```
     min 25\\% median 75\\%   max   mean  std 0s NAs   n
v.1.   2  1800   2400  3000 36000 2640.2 2530  0   0 810
```

```
Ls1[, yagiValue := yagiM * 1900]
Ls1[, yagiLValue := yagiL * 1900 * .5]
```

Total livestock value.

```
Ls1[, totalLivestockValue := cowValue + calfValue + yagiValue +
        cowLValue + calfLValue + yagiLValue]
setkey(Ls1, hhid)
```

Rd2.

```
Ls2 <- a2b(Ls2, NA, 0)
Ls2[, livestockValue := s17a_3 * s17a_4]
Ls2[grepl("sh", s17a_2), livestockValue := livestockValue * .5]
Ls2[grepl("0", s17a_2), livestockValue := 0]
Ls2[, livestockValue := sum(livestockValue, na.rm = T), by = hhid]
Ls2[, livestockSoldValue := s17a_9]
Ls2[, livestockSoldValue := sum(livestockSoldValue, na.rm = T),
        by = hhid]
Ls2[, livestockDCValue := (s17a_6 + s17a_7) * s17a_4]
Ls2[, livestockDCValue := sum(livestockDCValue, na.rm = T),
        by = hhid]
Ls2[, totalLivestockValue := sum(livestockValue + livestockSoldValue + livestockDCValue),
Ls2 <- Ls2[!duplicated(Ls2[, hhid]), ]
#Ls2[grepl("a", s17a_code) & s17a_8 > 0, .(s17a_8, s17a_9)]
```

Rd3.

```
Ls3 ← a2b(Ls3, NA, 0)
Ls3[, livestockValue := s17a_3 * s17a_4]
Ls3[grepl("sh", s17a_2), livestockValue := livestockValue * .5]
Ls3[grepl("0", s17a_2), livestockValue := 0]
Ls3[, livestockValue := sum(livestockValue, na.rm = T), by = hhid]
Ls3[, livestockSoldValue := s17a_9]
Ls3[, livestockSoldValue := sum(livestockSoldValue, na.rm = T),
        by = hhid]
Ls3[, livestockDCValue := (s17a_6 + s17a_7) * s17a_4]
Ls3[, livestockDCValue := sum(livestockDCValue, na.rm = T),
        by = hhid]
Ls3[, totalLivestockValue := sum(livestockValue + livestockSoldValue + livestockDCValue),
Ls3 ← Ls3[!duplicated(Ls3[, hhid]), ]
```

Merge.

```
ls ← rbind(cbind(rd = 1, Ls1[, .(hhid, totalLivestockValue)]),
        cbind(rd = 2, Ls2[, .(hhid, totalLivestockValue)]),
        cbind(rd = 3, Ls3[, .(hhid, totalLivestockValue)]))
ls ← ls[!duplicated(ls), ]
ls[, totalLivestockValue := totalLivestockValue/1000]
setkey(ls, hhid, rd); setkey(tr11, hhid, rd)
lst ← tr01[ls]
```

```
lstk.ss ← subset(lst, !is.na(gid) & !is.na(elapsed))
setkey(lstk.ss, rd, gid, assignment)
lstk.ss[, avgElapsed := mean(elapsed, na.rm = T), by = c("rd", "gid", "assignment")]
lstk.ss[, avgElapsed0 := avgElapsed[1], by = c("rd", "gid")]
lstk.ss[, avgElapsed1 := avgElapsed[.N], by = c("rd", "gid")]
lstk.ss[, avgLstkValue := mean(totalLivestockValue, na.rm = T),
        by = c("rd", "gid", "assignment")]
lstk.ss[, avgLstkValue0 := avgLstkValue[1], by = c("rd", "gid")]
lstk.ss[, avgLstkValue1 := avgLstkValue[.N], by = c("rd", "gid")]
lstk.ss[gid == 70204& rd == 1, .(rd, gid, assignment, avgElapsed,
        avgElapsed0, avgElapsed1, avgLstkValue, avgLstkValue0, avgLstkValue1) ]
```

```
    rd   gid assignment avgElapsed avgElapsed0 avgElapsed1 avgLstkValue
 1:  1 70204    control    561.286     561.286       837.1        3.350
 2:  1 70204    control    561.286     561.286       837.1        3.350
 3:  1 70204    control    561.286     561.286       837.1        3.350
 4:  1 70204    control    561.286     561.286       837.1        3.350
 5:  1 70204    control    561.286     561.286       837.1        3.350
 6:  1 70204    control    561.286     561.286       837.1        3.350
 7:  1 70204    control    561.286     561.286       837.1        3.350
 8:  1 70204    treated    837.100     561.286       837.1        9.105
 9:  1 70204    treated    837.100     561.286       837.1        9.105
10:  1 70204    treated    837.100     561.286       837.1        9.105
11:  1 70204    treated    837.100     561.286       837.1        9.105
12:  1 70204    treated    837.100     561.286       837.1        9.105
13:  1 70204    treated    837.100     561.286       837.1        9.105
14:  1 70204    treated    837.100     561.286       837.1        9.105
15:  1 70204    treated    837.100     561.286       837.1        9.105
16:  1 70204    treated    837.100     561.286       837.1        9.105
17:  1 70204    treated    837.100     561.286       837.1        9.105
    avgLstkValue0 avgLstkValue1
 1:          3.35         9.105
 2:          3.35         9.105
 3:          3.35         9.105
 4:          3.35         9.105
```

```
 5:           3.35         9.105
 6:           3.35         9.105
 7:           3.35         9.105
 8:           3.35         9.105
 9:           3.35         9.105
10:           3.35         9.105
11:           3.35         9.105
12:           3.35         9.105
13:           3.35         9.105
14:           3.35         9.105
15:           3.35         9.105
16:           3.35         9.105
17:           3.35         9.105
```

```
lstk.ss[, avgDiffElapsed := avgElapsed1 − avgElapsed0]
lstk.ss[, avgDiffLstkValue := avgLstkValue1 − avgLstkValue0]
setkey(lstk.ss, rd, gid, assignment)
dim(lstk.sss ← lstk.ss[!duplicated(lstk.ss[, .(rd, gid, assignment)]), ])
```

```
[1] 540  28
```

We compare the effects of treatment exposure differences within the same group.

```
library(ggplot2)
ggplot(data = lstk.sss, aes(x = avgDiffElapsed, y = avgDiffLstkValue)) +
        geom_point(aes(colour = arm, shape = arm), size = .05) +
        scale_shape(solid = F) + scale_y_continuous() +
        xlab("difference in elapsed days") + ylab("difference in mean livestock value (Tk
        labs(fill = "arm") + facet_grid( ~ rd) +
#       stat_smooth(method = "loess", size = .2, n = 150) +
        geom_smooth(method = "loess", size = .2) +
        geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed", size = .2) +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

```
library(ggplot2)
ggplot(data = lstk.sss, aes(x = avgDiffElapsed, y = avgDiffLstkValue)) +
        geom_point(aes(colour = arm, shape = arm), size = .05) +
        scale_shape(solid = F) +
        xlab("difference in elapsed days") + ylab("difference in mean livestock value (Tk
        labs(fill = "arm") + facet_grid(arm ~ rd) +
#       stat_smooth(method = "loess", size = .2, n = 150) +
        geom_smooth(method = "loess", size = .2) +
        geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed", size = .2) +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
```

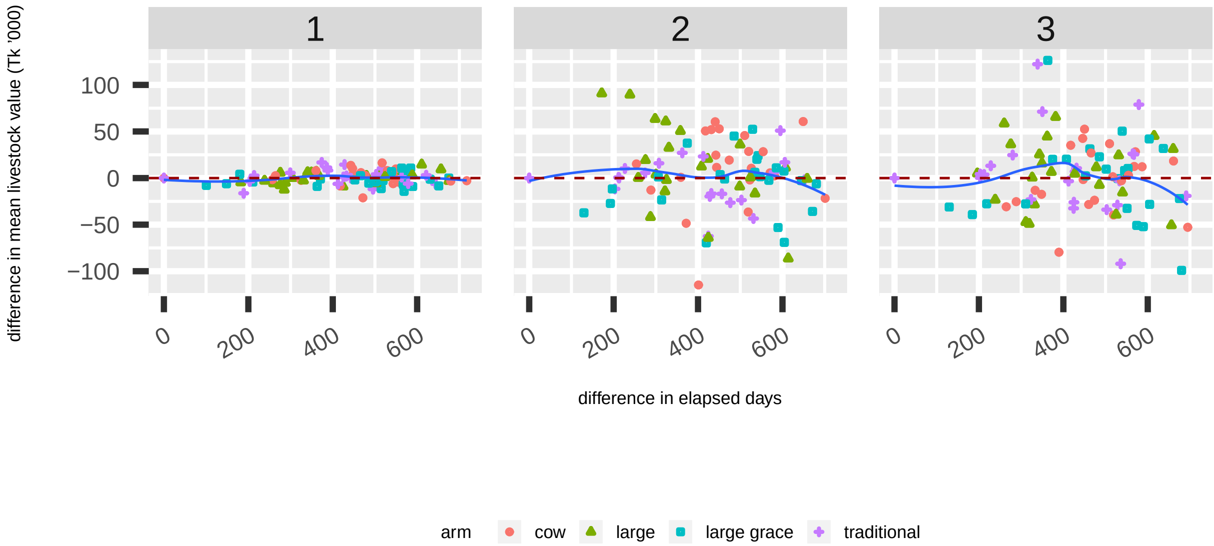

Figure 11 Livestock by elapsed days from disbursement within a group

```
                        legend.text = element_text(size=rel(.25)),
                        legend.position = "bottom",
                        legend.title = element_text(size = rel(.25)),
                        legend.key = element_rect(size = rel(.25)),
                        legend.key.size = unit(.15, "cm"),
                        strip.text = element_text(size=rel(.5)),
                        strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                        strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

Add assets and livestock.

```
al.ss ← merge(asset.ss, lstk.ss,
        by = c("rd", "gid", "hhid", "assignment", "arm", "elapsed"), all = T)
al.ss[is.na(assetNPV), assetNPV := 0]
al.ss[is.na(totalLivestockValue), totalLivestockValue := 0]
al.ss[, val := (assetNPV/1000 + totalLivestockValue)]
setkey(al.ss, rd, gid, assignment)
al.ss[, avgElapsed := mean(elapsed, na.rm = T), by = c("rd", "gid", "assignment")]
al.ss[, avgElapsed0 := avgElapsed[1], by = c("rd", "gid")]
al.ss[, avgElapsed1 := avgElapsed[.N], by = c("rd", "gid")]
al.ss[, avgVal := mean(val, na.rm = T), by = c("rd", "gid", "assignment")]
al.ss[, avgVal0 := avgVal[1], by = c("rd", "gid")]
al.ss[, avgVal1 := avgVal[.N], by = c("rd", "gid")]
unique(al.ss[gid == 70101 & rd == 1, .(avgElapsed0, avgElapsed1)])
```

```
   avgElapsed0 avgElapsed1
1:       688.3         892
```

```
al.ss[, avgDiffElapsed := avgElapsed1 - avgElapsed0]
al.ss[, avgDiffVal := avgVal1 - avgVal0]
dim(al.sss ← al.ss[!duplicated(al.ss[, .(rd, gid, assignment)]), ])
```

```
[1] 543  80
```

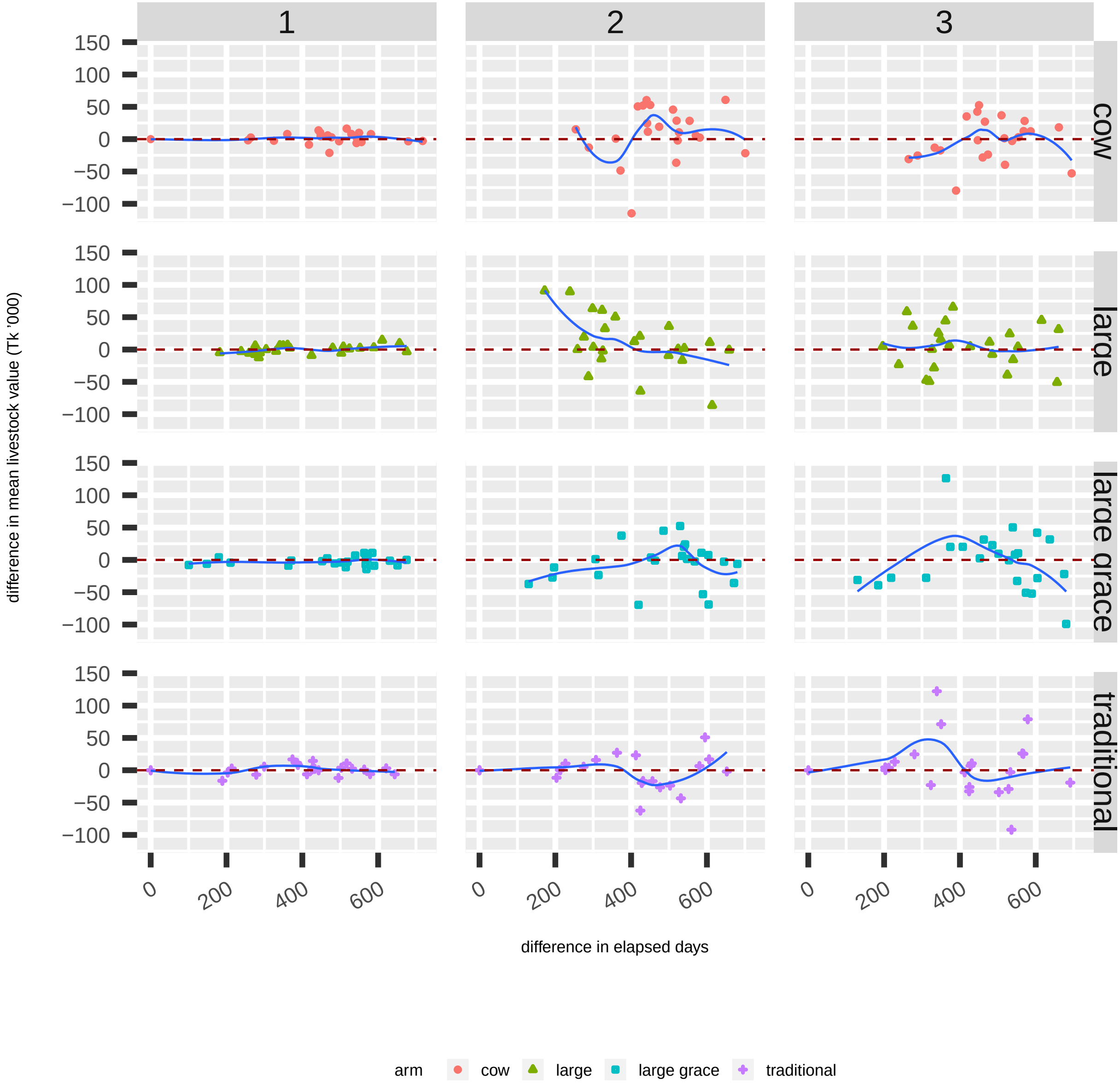

Figure 12 Livestock by elapsed days from disbursement, within a group

```
library(ggplot2)
ggplot(data = al.sss , aes(x = avgDiffElapsed , y = avgDiffVal)) +
          geom_point(aes(colour = arm, shape = arm), size = .05) +
          scale_shape(solid = F) +
          xlab("difference in elapsed days") + ylab("difference in mean value (Tk '000)") +
          labs(fill = "arm") + facet_grid(. ~ rd) +
#         stat_smooth(method = "loess", size = .2, n = 150) +
          geom_smooth(method = "loess", size = .2) +
          geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed", size = .2) +
          theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                    axis.title.x = element_text(size = rel(.25), angle = 0),
                    axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                    axis.text.y = element_text(size = rel(.5), angle = 0),
                    legend.text = element_text(size=rel(.25)),
                    legend.position = "bottom",
```

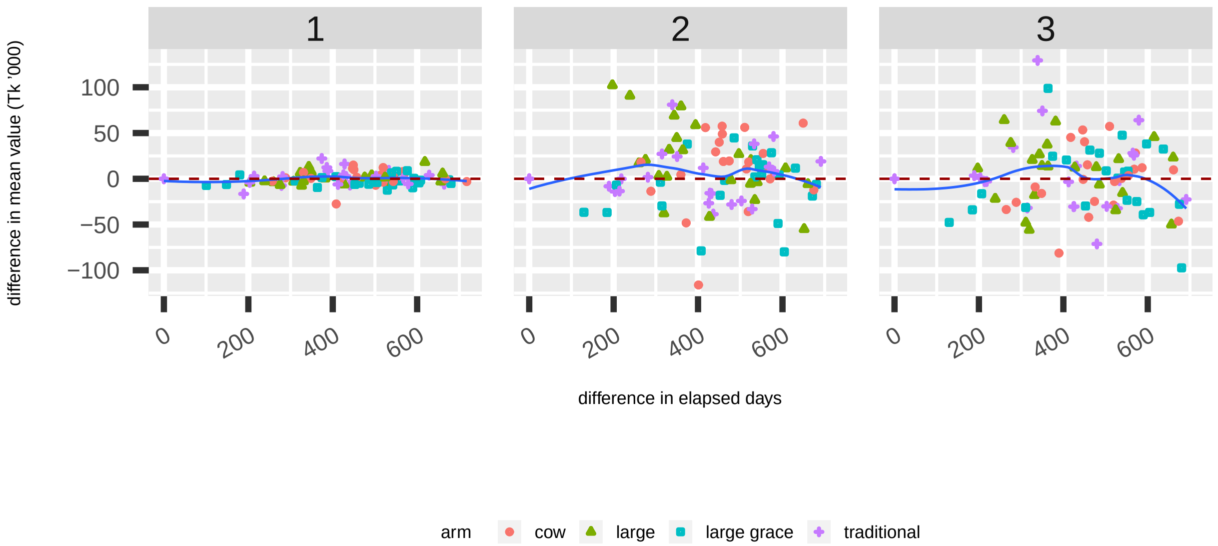

Figure 13 Total assets by elapsed days from disbursement within a group

```
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

```
library(ggplot2)
ggplot(data = al.sss, aes(x = avgDiffElapsed, y = avgDiffVal)) +
        geom_point(aes(colour = arm, shape = arm), size = .05) +
        scale_shape(solid = F) +
        xlab("difference in elapsed days") + ylab("difference in mean value (Tk 1000)") +
        labs(fill = "arm") + facet_grid(arm ~ rd) +
#       stat_smooth(method = "loess", size = .2, n = 150) +
        geom_smooth(method = "loess", size = .2) +
        geom_hline(aes(yintercept = 0), colour="#990000", linetype="dashed", size = .2) +
        theme(axis.title.y = element_text(size = rel(.25), angle = 90),
                axis.title.x = element_text(size = rel(.25), angle = 0),
                axis.text.x = element_text(size = rel(.5), angle = 30, hjust = 1),
                axis.text.y = element_text(size = rel(.5), angle = 0),
                legend.text = element_text(size=rel(.25)),
                legend.position = "bottom",
                legend.title = element_text(size = rel(.25)),
                legend.key = element_rect(size = rel(.25)),
                legend.key.size = unit(.15, "cm"),
                strip.text = element_text(size=rel(.5)),
                strip.text.x = element_text(margin = margin(.05, 0, .05, 0, "cm")),
                strip.text.y = element_text(margin = margin(.05, 0, .05, 0, "cm")))
```

Regressions. First, get roster files to obtain hh background.

```
setwd(pathsource.mar)
foldername ← list.dirs(path = ".", recursive = T, full.names = T)
foldername ← foldername[grepl("add|ori", foldername)]
fn1 ← unique(list.files(path = foldername, pattern = ".prn$",
```

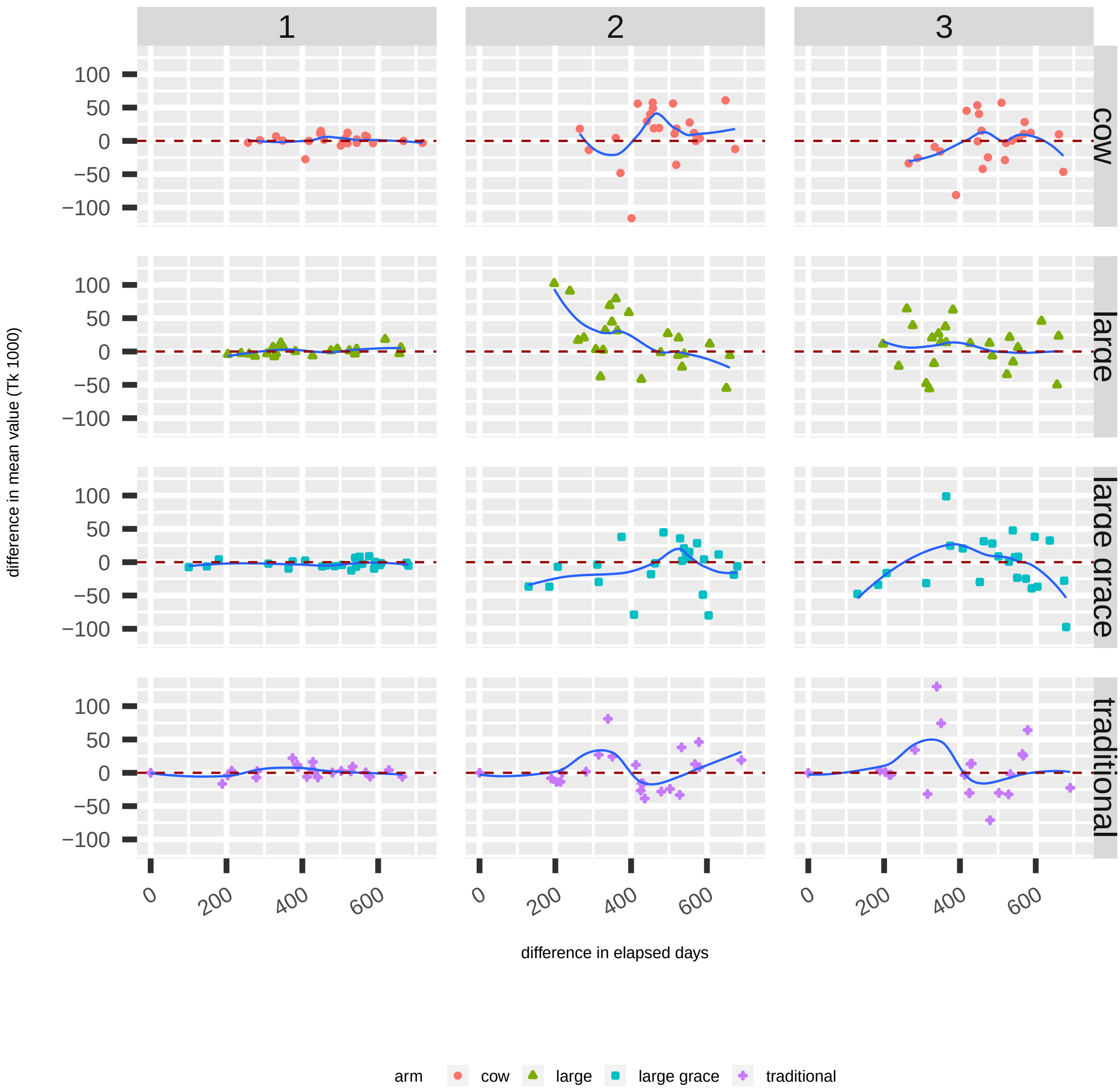

Figure 14 Total assets by elapsed days from disbursement, within a group

```
          recursive = T, full.names = T))
fn.ros ← grepout("s1.p1|Se.*01", fn1)
Ro = lapply(fn.ros, fread, integer64 = "double")
ro1 ← rbindlist(Ro, fill = T, use.names = T)
ro1 ← ro1[!duplicated(ro1[, .(hhid, mid, memname)]), ]
ro1[, numAdults := sum(age_1 > 15 & age_1 ≤ 60, na.rm = T), by = hhid]
ro1[, numChildren := sum(age_1 ≤ 15, na.rm = T), by = hhid]
ro1[, numElderly := sum(age_1 > 60, na.rm = T), by = hhid]
ro1[, numDisabled := sum(grepl("Y|1", disability)), by = hhid]
ro1[, numMale := sum(grepl("M|1", sex)), by = hhid]
ro1[, numLiterate := sum(grepl("Can.*and", literacy) | grepl("4", lliteracy)), by = hhid]
ro1[, headLiterate :=
          (grepl("Can.*and", literacy)  | grepl("4", lliteracy)) & grepl("He|1", rel_hhh),
          by = hhid]
ro1[, numLiterateMale :=
```

```
        sum((grepl("Can.∗and", literacy)  | grepl("4", lliteracy)) & grepl("M|1", sex)),
        by = hhid]
ro ← ro1[, .(hhid, numAdults, numChildren, numElderly,
        numDisabled, numMale, numLiterate, numLiterateMale, headLiterate)]
ro ← ro[!duplicated(ro), ]
setwd(pathsave)
write.tablev(ro, "rd1_roster_summary.prn")
```

Summarise at cluster level.

```
tr3 ← tr[, .(gid, hhid)]
setkey(ro1, hhid); setkey(tr3, hhid);
ros ← tr3[ro1]
ros[, size := .N, by = gid]
ros[, ratioAdults := sum(age_1 > 15 & age_1 ≤ 60, na.rm = T)/size, by = gid]
ros[, ratioChildren := sum(age_1 ≤ 15, na.rm = T)/size, by = gid]
ros[, ratioElderly := sum(age_1 > 60, na.rm = T)/size, by = gid]
ros[, ratioDisabled := sum(grepl("Y|1", disability))/size, by = gid]
ros[, ratioMale := sum(grepl("M|1", sex))/size, by = gid]
ros[, ratioLiterate := sum(grepl("Can.∗and", literacy) | grepl("4", lliteracy))/size, by =
ros[, ratioHeadLiterate :=
        sum((grepl("Can.∗and", literacy)  | grepl("4", lliteracy)) &
        grepl("He|1", rel_hhh))/size, by = gid]
ros[, ratioLiterateMale :=
        sum((grepl("Can.∗and", literacy)  | grepl("4", lliteracy)) & grepl("M|1", sex))/si
        by = gid]
ro2 ← ros[, .(gid, size, ratioAdults, ratioChildren, ratioElderly,
        ratioDisabled, ratioMale, ratioLiterate, ratioLiterateMale, ratioHeadLiterate)]
ro2 ← ro2[!duplicated(ro2[, gid]), ]
ro2 ← ro2[!is.na(gid), ]
```

Merge with asset data.

```
setkey(al.sss, gid, rd); setkey(ro2, gid)
alr.sss ← ro2[al.sss]
dim(alr.sss ← alr.sss[!duplicated(alr.sss[, .(rd, gid)]), ])
```

```
[1] 276  89
```

```
setkey(alr.sss, gid, rd)
alr.sss[, exist := .N, by = gid]
dim(alr.sss ← alr.sss[exist == 3, ])
```

```
[1] 276  90
```

```
destat.alr ← destat(alr.sss[, .(elapsed,
        size, ratioChildren, ratioAdults, ratioDisabled, ratioMale, ratioLiterate,
        ratioLiterateMale, ratioHeadLiterate, avgDiffElapsed,
        avgDiffVal, avgVal, avgVal1, avgVal0, avgElapsed, avgElapsed1, avgElapsed0)])
destat.alr ← cbind(rownames(destat.alr), destat.alr)
setwd(pathsave)
ltab.alr ← latextab(destat.alr, headercolor = "blue!10",
        alternatecolor = "gray90", delimiterline = NULL,
        hleft = c("\\footnotesize\\hfill", rep("\\footnotesize\\hfil$", ncol(destat.alr)−
        hcenter = c("2", rep("1.0", ncol(destat.alr)−1)),
        hright = c("", rep("$", ncol(destat.alr)−1)))
write.tablev(ltab.alr, "destat_alr_sss.tex", colnamestrue = F)
dalr.sss1 ← cbind(d.rd = 1, alr.sss[rd==1, .(gid, assignment, arm, elapsed,
```

```
        size, ratioChildren, ratioAdults, ratioDisabled, ratioMale, ratioLiterate,
        ratioLiterateMale, ratioHeadLiterate, avgDiffElapsed)],
        alr.sss[rd==2, .(avgDiffVal, avgDiffLstkValue, avgVal, avgVal1, avgVal0, avgElapse
        alr.sss[rd==1, .(avgDiffVal, avgDiffLstkValue, avgVal, avgVal1, avgVal0, avgElapse
dalr.sss2 ← cbind(d.rd = 2, alr.sss[rd==1, .(gid, assignment, arm, elapsed,
        size, ratioChildren, ratioAdults, ratioDisabled, ratioMale, ratioLiterate,
        ratioLiterateMale, ratioHeadLiterate, avgDiffElapsed)],
        alr.sss[rd==3, .(avgDiffVal, avgDiffLstkValue, avgVal, avgVal1, avgVal0, avgElapse
        alr.sss[rd==2, .(avgDiffVal, avgDiffLstkValue, avgVal, avgVal1, avgVal0, avgElapse
dalr.sss ← rbind(dalr.sss1, dalr.sss2)

l1 ← glm(avgDiffVal ~ avgDiffElapsed, data = dalr.sss)
l2 ← glm(avgDiffVal ~ avgDiffElapsed:arm, data = dalr.sss)
l3 ← glm(avgDiffVal ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale, data = dalr.sss)
l4 ← glm(avgDiffVal ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale +
        ratioLiterate + ratioLiterateMale + ratioHeadLiterate, data = dalr.sss)
linprob ← list(l1, l2, l3, l4)
linest ← lapply(linprob, clx.regobj, Cluster = "gid")
linest ← lapply(linest, function(x) x[, -3])
linest ← tabs2latex(linest)
R2 ← round(asn(lapply(linprob,
        function(x) 1-crossprod(summary(x)$deviance.res)/summary(x)$null.dev)), 3)
en ← asn(lapply(linprob, function(x) length(x$y)))
rn ← rownames(linest)
rn ← gsub("arm|^se.*", "", rn)
rn ← gsub(":", " * ", rn)
ltab ← rbind(as.matrix(cbind(rn, linest)), c("$R^{2}$", R2),
        c("$n$", en))
write.tablev(latextab(ltab, delimiterline = NULL, alternatecolor2 = "gray90",
        hleft = c("\\footnotesize", rep("\\scriptsize\\hfil$", ncol(ltab)-1)),
        hcenter = c(3.5, rep(1.5, ncol(ltab)-1)),
        hright = c("\\hfill", rep("$", ncol(ltab)-1)),
        adjustlineskip = "-.4ex"),
        paste0(pathsave, "asset_regression_alr_sss.tex"), colnamestrue = F)

ll1 ← glm(avgDiffLstkValue ~ avgDiffElapsed, data = dalr.sss)
ll2 ← glm(avgDiffLstkValue ~ avgDiffElapsed:arm, data = dalr.sss)
ll3 ← glm(avgDiffLstkValue ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale, data = dalr.sss)
ll4 ← glm(avgDiffLstkValue ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale +
        ratioLiterate + ratioLiterateMale + ratioHeadLiterate, data = dalr.sss)
llinprob ← list(ll1, ll2, ll3, ll4)
llinest ← lapply(llinprob, clx.regobj, Cluster = "gid")
llinest ← lapply(llinest, function(x) x[, -3])
llinest ← tabs2latex(llinest)
R2 ← round(asn(lapply(llinprob,
        function(x) 1-crossprod(summary(x)$deviance.res)/summary(x)$null.dev)), 3)
en ← asn(lapply(llinprob, function(x) length(x$y)))
rn ← rownames(llinest)
rn ← gsub("arm|^se.*", "", rn)
rn ← gsub(":", " * ", rn)
lltab ← rbind(as.matrix(cbind(rn, llinest)), c("$R^{2}$", R2),
        c("$n$", en))
```

Table 2: Descriptive statistics of asset regression data

	min	25%	median	75%	max	mean	std	0s	NAs	n
elapsed	49	296	352	556	892	400.8	204.6	0	0	276
size	65	82.8	89	97	171	91.8	17	0	0	276
ratioChildren	0.3	0.4	0.4	0.5	0.6	0.4	0.1	0	0	276
ratioAdults	0.4	0.5	0.6	0.6	0.7	0.6	0.1	0	0	276
ratioDisabled	0	0	0	0	0	0	0	168	0	276
ratioMale	0.4	0.5	0.5	0.5	0.6	0.5	0	0	0	276
ratioLiterate	0	0.2	0.3	0.4	0.5	0.3	0.1	0	0	276
ratioLiterateMale	0	0.1	0.2	0.2	0.3	0.2	0.1	0	0	276
ratioHeadLiterate	0	0	0	0	0.1	0	0	51	0	276
avgDiffElapsed	0	331.6	452.3	543.5	717.4	431.7	156.3	9	0	276
avgDiffVal	−116	−7.1	1	14.3	129.4	3	30.7	9	0	276
avgVal	1.2	15.2	43.2	87.6	192.4	56.4	48.3	0	0	276
avgVal1	0.7	15.3	54.8	88.8	230.4	59.4	48.4	0	0	276
avgVal0	1.2	15.2	43.2	87.6	192.4	56.4	48.3	0	0	276
avgElapsed	122.6	267.9	346.3	477.5	892	374.5	145.7	0	0	276
avgElapsed1	544.3	729	834.8	857.3	899	806.2	71.8	0	0	276
avgElapsed0	122.6	267.9	346.3	477.5	892	374.5	145.7	0	0	276

```
write.tablev(latextab(lltab, delimiterline = NULL, alternatecolor2 = "gray90",
        hleft = c("\\footnotesize", rep("\\scriptsize\\hfil$", ncol(lltab)-1)),
        hcenter = c(3.5, rep(1.5, ncol(lltab)-1)),
        hright = c("\\hfill", rep("$", ncol(lltab)-1)),
        adjustlineskip = "-.4ex"),
        paste0(pathsave, "livestock_regression_alr_sss.tex"), colnamestrue = F)
```

References

Egger, Peter H. and Maximilian von Ehrlich, "Generalized propensity scores for multiple continuous treatment variables," *Economics Letters*, 2013, *119* (1), 32 – 34.

Hirano, Keisuke and Guido W. Imbens, *The Propensity Score with Continuous Treatments*, John Wiley & Sons, Ltd,

Imai, Kosuke and David A van Dyk, "Causal Inference With General Treatment Regimes," *Journal of the American Statistical Association*, 2004, *99* (467), 854–866.

Imbens, Guido W., "The role of the propensity score in estimating dose-response functions," *Biometrika*, 2000, *87* (3), 706.

Kluve, Jochen, Hilmar Schneider, Arne Uhlendorff, and Zhong Zhao, "Evaluating continuous training programmes by using the generalized propensity score," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2012, *175* (2), 587–617.

Table 3: DID estimates of asset impacts

rn	(1)	(2)	(3)	(4)
(Intercept)	3.511	3.127	−177.737	−156.113
	(4.794)	(4.744)	(162.074)	(162.405)
avgDiffElapsed	−0.006			
	(0.010)			
avgDiffElapsed * cow		−0.008	−0.006	−0.003
		(0.011)	(0.010)	(0.011)
avgDiffElapsed * large		−0.003	−0.003	−0.003
		(0.012)	(0.011)	(0.011)
avgDiffElapsed * large grace		−0.006	−0.009	−0.005
		(0.013)	(0.013)	(0.013)
avgDiffElapsed * traditional		−0.004	−0.010	−0.008
		(0.014)	(0.014)	(0.015)
size			−0.027	−0.007
			(0.107)	(0.104)
ratioChildren			150.652	144.698
			(165.090)	(161.753)
ratioAdults			202.275	191.880
			(165.389)	(160.061)
ratioDisabled			−374.066**	−364.441**
			(173.656)	(171.045)
ratioMale			17.408	−28.147
			(33.939)	(50.547)
ratioLiterate				−33.128
				(36.090)
ratioLiterateMale				105.570
				(77.780)
ratioHeadLiterate				−6.597
				(110.909)
R^2	0.001	0.001	0.022	0.028
n	184	184	184	184

Notes: 1. Difference-in-differences estimates of asset accumulation against elapsed days.

2. large, large grace, cow are all time invariant and are interacted with a trend term.

3. $*$, $**$, $***$ indicate significance levels at 10%, 5%, 1%, respectively.

Table 4: DID estimates of livestock impacts

rn	(1)	(2)	(3)	(4)
(Intercept)	3.760	3.312	−132.813	−107.738
	(5.113)	(5.221)	(243.731)	(235.304)
avgDiffElapsed	−0.006			
	(0.012)			
avgDiffElapsed * cow		−0.009	−0.006	− 0.000
		(0.012)	(0.013)	(0.014)
avgDiffElapsed * large		0.002	0.001	− 0.000
		(0.014)	(0.013)	(0.015)
avgDiffElapsed * large grace		−0.005	−0.010	−0.007
		(0.014)	(0.016)	(0.017)
avgDiffElapsed * traditional		−0.009	−0.022	−0.020
		(0.019)	(0.019)	(0.021)
size			−0.004	0.043
			(0.113)	(0.118)
ratioChildren			61.527	63.033
			(241.252)	(231.535)
ratioAdults			163.485	171.256
			(249.577)	(236.432)
ratioDisabled			−392.670**	−413.142**
			(199.920)	(210.322)
ratioMale			45.227	−37.036
			(42.598)	(54.860)
ratioLiterate				−61.047
				(42.057)
ratioLiterateMale				174.673**
				(85.504)
ratioHeadLiterate				−115.088
				(125.939)
R^2	0.001	0.003	0.033	0.045
n	141	141	141	141

Notes: 1. Difference-in-differences estimates of asset accumulation against elapsed days.

2. large, large grace, cow are all time invariant and are interacted with a trend term.

3. $*$, $**$, $***$ indicate significance levels at 10%, 5%, 1%, respectively.