Comparing outcomes between groups

April 6, 2017
17:01

Seiro Ito

Contents

There are a few key variables.

disbursed If a subject received a loan. This is time-variant. Most of subjects are disbursed=F in rd 1 but T in later rds.

receivedCredit If a subject received a loan in any of the round. This is time-invariant.

assignment The original treatment assignment. This is time-invariant. 10 of 20 group members are assigned to control, remaining 10 are treated. There are subjects who dropped out before the assignment was determined that we label as "quit before assinged". There are also subject who refused to get treated, so staying in the treated group but not receiving a credit.

arm Arms are "traditional" (reference), "large," "large grace," "cow," and "lost before assigned." Some groups/individuals rejected before an arm is assigned, some rejected after an arm is assigned.

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 222 cases of rejections who are group rejections (140), lost to flood (80), and old members (2).

```
setwd(pathsave)
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, c("traditional", "large", "large grace", "cow",
        "group quit after randomise", "lost to flood"))]
tr[, assignment := factor(assignment,
        levels = c("quit after randomise", "control", "treated"))]
tr[, memstatus := factor(memstatus)]
tr0 ← tr[, .(gid, hhid, memstatus, assignment, arm, accept,
        receivedCredit, elapsed, daysFromStart)]
table0(tr0[, .(memstatus, assignment)])
```

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

```
#table0(idt[grepl("ye", accept) & grepl("qu", assignment), .(memstatus, arm)])
table0(tr0[grepl("old", memstatus) & is.na(assignment), .(accept, arm)])
```

```
                      arm
accept                 cow
  individual rejection   2
```

These 2 subjects who accepted but NA in receivedCredit, 7137316, 7137317 are under cow arm.

```
table0(tr0[, .(accept, assignment)])
```

```
                      assignment
accept                 control treated <NA>
  group rejection           10      10  140
  individual rejection      69      90    2
  yes                      899     899   80
```

There are subjects who accepted but quit before individual randomisation. This is not impossible but looks odd.

```
table0(tr0[grepl("ye", accept) & is.na(assignment), .(memstatus, arm)])
```

```
                      arm
memstatus              lost to flood
  group rejection                  0
  individual rejection             0
  lost to flood                   80
  new group                        0
  old                              0
  replacement                      0
```

There are many subjects rejected cows.

```
table0(tr0[, .(accept, arm)])
```

	arm					
accept	traditional	large	large grace	cow	lost to flood	
group rejection	80	40	40	0	0	
individual rejection	53	12	22	74	0	
yes	480	440	440	438	80	

Among indiviual rejecters who rejected cows, the proportion is equally distributed among the treated and the control.

```
table0(tr0[grepl("ind", accept), .(arm, assignment)])
```

	assignment		
arm	control	treated	<NA>
traditional	22	31	0
large	3	9	0
large grace	9	13	0
cow	35	37	2
group quit after randomise	0	0	0
lost to flood	0	0	0

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

	assignment		
arm	control	treated	<NA>
traditional	262	271	80
large	223	229	40
large grace	239	243	20
cow	254	256	2
group quit after randomise	0	0	0
lost to flood	0	0	80

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

	arm					
memstatus	traditional	large	large grace	cow	lost to flood	
group rejection	80	40	40	0	0	
individual rejection	53	12	22	72	0	
lost to flood	0	0	0	0	80	
new group	200	80	80	60	0	
old	227	348	338	308	0	
replacement	53	12	22	72	0	

```
table0(tr0[is.na(elapsed), .(arm, assignment)])
```

	assignment		
arm	control	treated	<NA>
traditional	96	31	80
large	3	9	40
large grace	9	13	20
cow	35	37	2
group quit after randomise	0	0	0
lost to flood	0	0	80

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

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

```
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.

```
setkey(indate, hhid, memstatus, receivedCredit)
setkey(tr0, hhid, memstatus, receivedCredit)
setkey(tr1, hhid, memstatus)
tr0 ← indate[tr0]
tr1 ← indate[tr1]
```

```
# reshape using shorter idvar vector works but longer (& unique) idvar vector does not.
any(duplicated(tr1[, .(gid, hhid, assignment, arm, memstatus)]))
```

```
[1] FALSE
```

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

If "quit before assigned," or "group rejection," or "individual rejection," or "lost to flood," then disbursed=F for all rounds.

```
tr1l[grepl("qui", assignment) | grepl("rej|lost", memstatus), disbursed := F]
table(tr1l[ ,.(rd, disbursed, assignment)], useNA = "ifany")
```

```
, , assignment = quit after randomise

   disbursed
rd  FALSE TRUE <NA>
  1     0    0    0
  2     0    0    0
  3     0    0    0

, , assignment = control

   disbursed
rd  FALSE TRUE <NA>
  1   978    0    0
  2   412  395  171
  3   173  779   26

, , assignment = treated

   disbursed
rd  FALSE TRUE <NA>
  1   999    0    0
  2   154  752   93
  3   127  852   20
```

```
, , assignment = NA

   disbursed
rd  FALSE TRUE <NA>
  1   222    0    0
  2   220    0    2
  3   220    0    2
```

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

```
, , memstatus = group rejection

   disbursed
rd  FALSE TRUE <NA>
  1   160    0    0
  2   160    0    0
  3   160    0    0

, , memstatus = individual rejection

   disbursed
rd  FALSE TRUE <NA>
  1   159    0    0
  2   159    0    0
  3   159    0    0

, , memstatus = lost to flood

   disbursed
rd  FALSE TRUE <NA>
  1    80    0    0
  2    80    0    0
  3    80    0    0

, , memstatus = new group

   disbursed
rd  FALSE TRUE <NA>
  1   420    0    0
  2    84  300   36
  3    37  378    5

, , memstatus = old

   disbursed
rd  FALSE TRUE <NA>
  1  1221    0    0
  2   265  756  200
  3    64 1117   40

, , memstatus = replacement

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

Who are disbursed=NA in rd 2 and assignment=control? Either missing in rd 2 or present but intDate or disburseDate is missing so we cannot see if the subject received a loan by the time of interview.

```
table(tr1l[rd == 2 & is.na(disbursed) & grepl("con", assignment), exist])
```

```
  1  12  13 123
 19   1  17 134
```

```
table0(is.na(tr1l[rd == 2 & is.na(disbursed) & grepl("con", assignment), intDate]) |
       tr1l[rd == 2 & is.na(disbursed) & grepl("con", assignment), disburseDate] == "")
```

```
TRUE
 171
```

assignment=treated but NA in disbursed: rd 2 missing intDate or disburseDate, rd 3 all attritions.

```
table(tr1l[rd == 3 & is.na(disbursed) & grepl("tr", assignment), exist])
```

```
 1 12
30 16
```

```
table(tr1l[rd == 2 & is.na(disbursed) & grepl("tr", assignment), exist])
```

```
  1  12  13 123
 30   2  27 205
```

```
table0(is.na(tr1l[rd == 2 & is.na(disbursed) & grepl("tr", assignment), intDate]) |
       tr1l[rd == 2 & is.na(disbursed) & grepl("tr", assignment), disburseDate] == "")
```

```
TRUE
 264
```

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])
```

```
named integer(0)
```

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

```
named integer(0)
```

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>
  55

[[2]]
<NA>
  23

[[3]]
<NA>
  19
```

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

```
[1] 2163   37 2055   58 2074   58
```

IV 3 meals per day

Three meals. In rd 1, we asked 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  1907  256
rd2  1201  854
rd3   980 1094
```

```
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
```

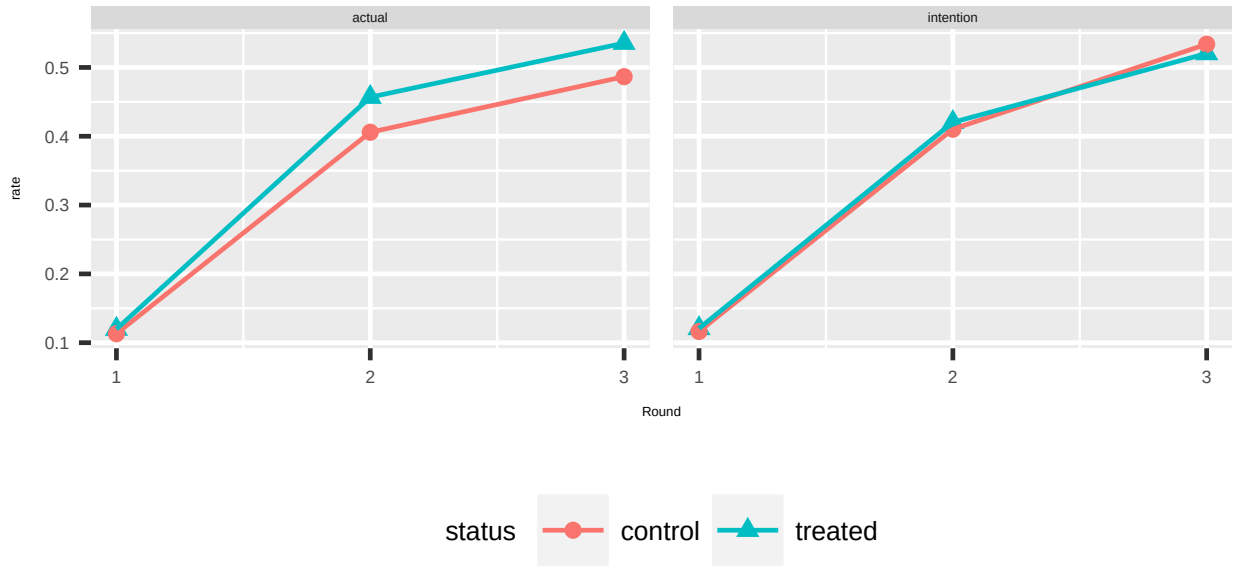

Figure 1 3 meals per day

	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:	1507	205	400	51	859	118	1048	138
3:	1000	683	201	169	565	393	636	461
4:	789	909	191	181	463	502	517	592

```
meal3DI ← data.frame(meal3DI)
colnames(meal3DI) ← 1:ncol(meal3DI)
meal3DI ← as.data.table(meal3DI[-1, ])
meal3DI ← as.data.table(sapply(meal3DI, asn))
#z.0 ← parse(text = "s8bq")
#sec3b[[1]][, zval := eval(z.0)]
```

D is actual treatment, I is intention to treat. Take a propotion.

```
setnames(meal3DI, colnames(meal3DI),
         paste0(repseq(c("D", "I"), 4), rep(repseq(c(1, 0), 2), 2), rep(c("F", "T"), 4)))
meal3DI[, rD0 := D0T/(D0T+D0F)]
meal3DI[, rD1 := D1T/(D1T+D1F)]
meal3DI[, rI0 := I0T/(I0T+I0F)]
meal3DI[, rI1 := I1T/(I1T+I1F)]
round(meal3DI[, .(rD0, rD1, rI0, rI1)], 2)
```

	rD0	rD1	rI0	rI1
1:	0.11	0.12	0.12	0.12
2:	0.46	0.41	0.42	0.41
3:	0.49	0.54	0.53	0.52

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 promissing.

Form data for regression. sec3b is a list of data tables for each rd. Merge each rd to form a wide format.

```
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] 2199   24
```

```
[1] 2199   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] 2197   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]),
        #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] 2197   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] 2024   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
2024
```

```
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")
```

```
Loading required package: sandwich
```

```
Loading required package: lmtest
```

```
Loading required package: zoo
```

```

Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
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$y)))
rn ← rownames(linest)
rn ← gsub("arm|assignment|^se.*", "", rn)
rn ← gsub("assigncredit", "treated * credit", rn)
rn ← gsub("disbursed", "credit", rn)
rn ← gsub(":", " * ", rn)
rn ← gsub("daysFromStart", "elapsed days * 100", rn)
ltab ← rbind(as.matrix(cbind(rn, linest)), c("$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.

TABLE 1: FD ESTIMATES OF THREE MEALS PER DAY, ROUND 2, 3

rn	(1)	(2)	(3)	(4)	(5)	(6)
(Intercept)	0.118*	0.143***	0.081	0.164***	0.142**	0.051
	(0.062)	(0.038)	(0.066)	(0.043)	(0.055)	(0.055)
large	−0.037		0.043		−0.033	0.099
	(0.099)		(0.095)		(0.097)	(0.085)
large grace	0.093		0.165		0.087	0.202**
	(0.090)		(0.101)		(0.084)	(0.095)
cow	−0.079		−0.051		−0.069	−0.008
	(0.101)		(0.101)		(0.098)	(0.098)
lost to flood	−0.043		−0.005			
	(0.133)		(0.134)			
treated		−0.032		−0.047		
		(0.024)		(0.079)		
drop out		−0.204*		−0.224*		
		(0.117)		(0.119)		
credit			0.147**	−0.056		0.143**
			(0.064)	(0.062)		(0.065)
large * credit			−0.366***			−0.421***
			(0.127)			(0.126)
large grace * credit			−0.228**			−0.263***
			(0.093)			(0.094)
cow * credit			−0.108			−0.122
			(0.134)			(0.136)
treated * credit				−0.012		
				(0.086)		
elapsed days * 100					−0.001	0.005
					(0.006)	(0.009)
R^2	0.011	0.008	0.023	0.01	0.01	0.029
n	2024	2024	1800	1800	1638	1489

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.

V 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.

Work hours.

VI 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"
```

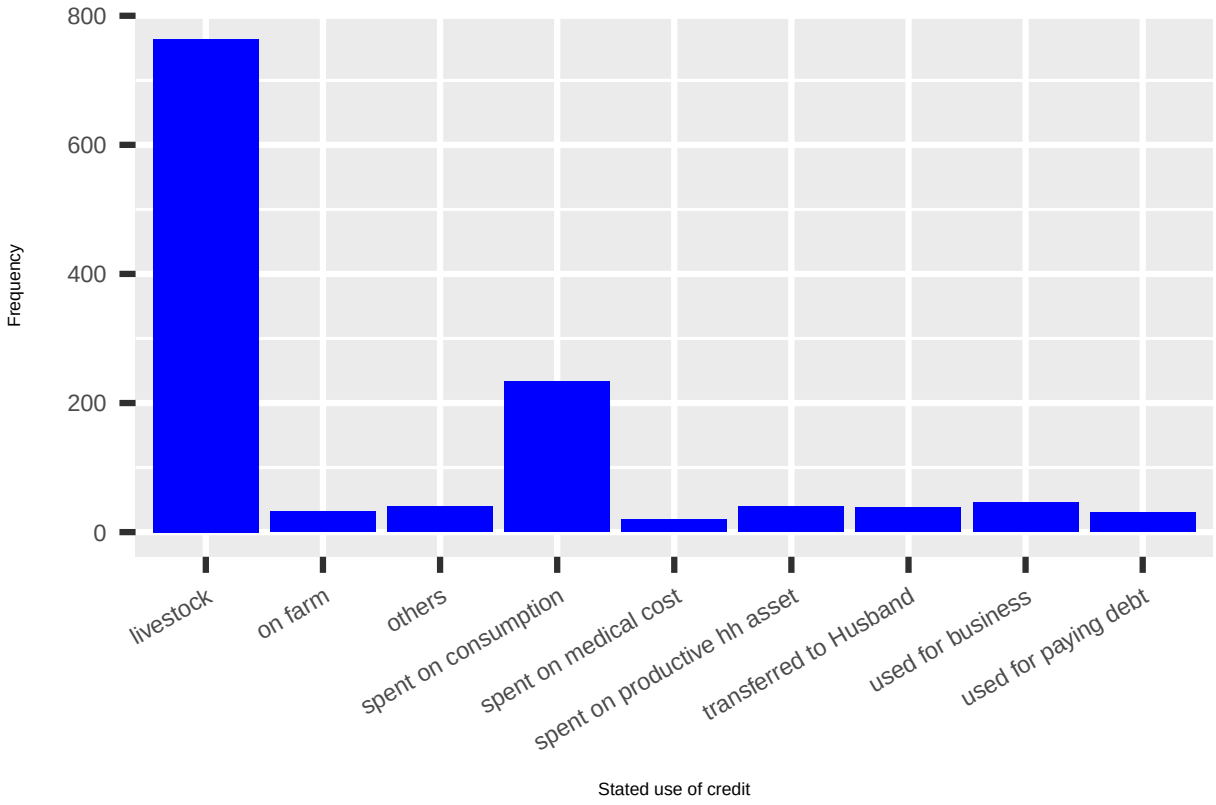

Figure 2 Stated use of credit in rd 2

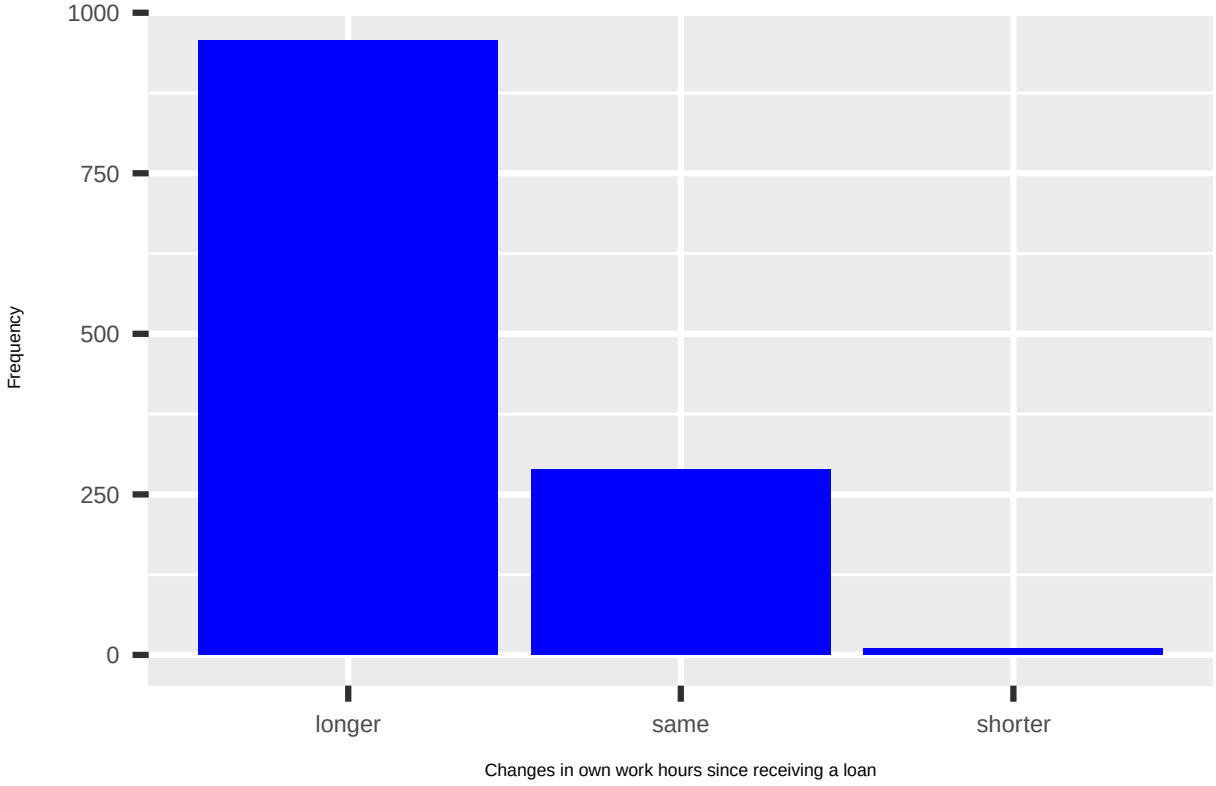

Figure 3 Work hours

```
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)))
bo11 ← reshape(bo1, direction = "long", idvar = "hhid",
        varying = grepout("\\.", colnames(bo1)))
setnames(bo11, "time", "lender")
setkey(bo11, hhid, lender)
bo11[, totalSum := sum(cashAmount, na.rm = T), by = hhid]
```

```
Warning in `[.data.table`(bo11, , `:=`(totalSum, sum(cashAmount, na.rm = T)), : Invalid .i
```

```
bo11[grepl("oth", lender), lender := "other NGO/MFI"]
bo11[grepl("rel", lender), lender := "friends, relatives"]
bo11[grepl("mo", lender), lender := "money lenders"]
Bo1 ← cbind(rd = 1, bo11)
```

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(bo11[, cashAmount], bo11[, 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)))
```

```
Warning: non-unique value when setting 'row.names': 'NA.1'
```

```
Error in `row.names<-.data.frame`(`*tmp*`, value = paste(d[, idvar], times[1L], : duplicat
```

```
setnames(tr01, "time", "rd")
```

```
Error in is.data.frame(x): object 'tr01' not found
```

```
setkey(Bo13, hhid, rd); setkey(tr01, hhid, rd)
```

```
Error in setkey(tr01, hhid, rd): object 'tr01' not found
```

```
Bot ← tr01[Bo13]
```

```
Error in eval(expr, envir, enclos): object 'tr01' not found
```

```
by(Bot[grepl("other|G", lender), cashAmount], Bot[grepl("other|G", lender), rd], destat)
```

```
Error in by(Bot[grepl("other|G", lender), cashAmount], Bot[grepl("other|G", : object 'Bot'
```

```
Bot[, combined.lender := lender]
```

```
Error in eval(expr, envir, enclos): object 'Bot' not found
```

```
Bot[grepl("oth|GU", lender), combined.lender := "NGO/MFI"]
```

```
Error in eval(expr, envir, enclos): object 'Bot' not found
```

```
Bot[grepl("shop", lender), combined.lender := "money lenders"]
```

```
Error in eval(expr, envir, enclos): object 'Bot' not found
```

```
setwd(pathsave)
write.tablev(Bot, "borrowing_rd_1-3.prn")
```

```
Error in is.data.frame(x): object 'Bot' not found
```

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),
                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")))
```

```
Error in subset(Bot, cashAmount > 0): object 'Bot' not found
```

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.

VII 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)
```

```
Error in setkey(tr01, hhid, rd): object 'tr01' not found
```

```
Asset01t ← tr01[Asset01]
```

```
Error in eval(expr, envir, enclos): object 'tr01' not found
```

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

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

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
asset.ss ← subset(asset, assetNPV > 0 & !is.na(receivedCredit))
```

```
Error in subset(asset, assetNPV > 0 & !is.na(receivedCredit)): object 'asset' not found
```

```
asset.cross ← tapply(asset.ss$assetNPV,
       list(rd = asset.ss$rd, receivedCredit=asset.ss$receivedCredit), median)
```

```
Error in tapply(asset.ss$assetNPV, list(rd = asset.ss$rd, receivedCredit = asset.ss$receiv
```

```
asset.cross2 ← tapply(asset.ss$assetNPV,
       list(rd = asset.ss$rd, receivedCredit=asset.ss$receivedCredit), mean)
```

```
Error in tapply(asset.ss$assetNPV, list(rd = asset.ss$rd, receivedCredit = asset.ss$receiv
```

```
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))
```

```
Error in cbind(vline.dat, median = c(asset.cross), mean = c(asset.cross2)): object 'asset.
```

```
library(ggplot2)
ggplot(data = subset(asset, assetNPV > 0 & !is.na(receivedCredit)),
        aes(x = assetNPV, fill = arm)) +
        geom_histogram(bins = 20) +
        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")))
```

```
Error in subset(asset, assetNPV > 0 & !is.na(receivedCredit)): object 'asset' not found
```

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)
```

```
Error in setkey(Asset01t, gid, hhid): object 'Asset01t' not found
```

```
# 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]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[, meanElapsedDaysOfGroup := mean(elapsed, na.rm = T), by = gid]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[, elaspsedGroupMedian := "early"]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[medianElapsedDaysOfGroup -
        median(medianElapsedDaysOfGroup, na.rm = T) <= 0,
        elaspsedGroupMedian := "late"]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[, elaspsedGroupMean := "early"]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[meanElapsedDaysOfGroup -
        mean(meanElapsedDaysOfGroup, na.rm = T) <= 0,
        elaspsedGroupMean := "late"]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[, elaspsedGroupMedian := factor(elaspsedGroupMedian)]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
Asset01t[, elaspsedGroupMean := factor(elaspsedGroupMean)]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
asset <- Asset01t[assetNumber == 1, ]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
asset.ss <- subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(disbursed))
```

```
Error in subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(disbursed)): object 'asset' not
```

```
asset.cross <- tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, disbursed=asset.ss$disbursed), median)
```

```
Error in tapply(asset.ss$assetNPV, list(rd = asset.ss$rd, disbursed = asset.ss$disbursed),
```

```
asset.cross2 <- tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, disbursed=asset.ss$disbursed), mean)
```

```
Error in tapply(asset.ss$assetNPV, list(rd = asset.ss$rd, disbursed = asset.ss$disbursed),
```

```
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))
```

```
Error in cbind(vline.dat, median = c(asset.cross), mean = c(asset.cross2)): object 'asset.
```

```
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")))
```

```
Error in ggplot(data = asset.ss, aes(x = assetNPV, fill = arm)): object 'asset.ss' not fou
```

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, ]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed) & receivedCredit)
```

```
Error in subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed) & : object 'asset' not
```

```
asset.cross ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, elaspsedGroupMedian = asset.ss$elaspsedGroupMedian), median
```

```
Error in tapply(asset.ss$assetNPV, list(rd = asset.ss$rd, elaspsedGroupMedian = asset.ss$e
```

```
asset.cross2 ← tapply(asset.ss$assetNPV,
        list(rd = asset.ss$rd, elaspsedGroupMedian = asset.ss$elaspsedGroupMedian), mean)
```

```
Error in tapply(asset.ss$assetNPV, list(rd = asset.ss$rd, elaspsedGroupMedian = asset.ss$e
```

```
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))
```

```
Error in cbind(vline.dat, median = c(asset.cross), mean = c(asset.cross2)): object 'asset.
```

```
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"),
                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")))
```

```
Error in ggplot(data = asset.ss, aes(x = assetNPV, fill = arm)): object 'asset.ss' not fou
```

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

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed) & receivedCredit)
```

```
Error in subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed) & : object 'asset' not
```

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

```
Error in ggplot(data = asset.ss, aes(x = elapsed, y = assetNPV)): object 'asset.ss' not fo
```

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, ]
```

```
Error in eval(expr, envir, enclos): object 'Asset01t' not found
```

```
asset.ss ← subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed))
```

```
Error in subset(asset, assetNPV > 0 & !is.na(gid) & !is.na(elapsed)): object 'asset' not f
```

```
setkey(asset.ss, rd, gid, assignment)
```

```
Error in setkey(asset.ss, rd, gid, assignment): object 'asset.ss' not found
```

```
asset.ss[, avgElapsed := mean(elapsed, na.rm = T), by = c("rd", "gid", "assignment")]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgElapsed0 := avgElapsed[1], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgElapsed1 := avgElapsed[.N], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[gid == 70650 & grepl("co", assignment), ]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

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

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgNPV := mean(assetNPV/1000, na.rm = T),
        by = c("rd", "gid", "assignment")]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgNPV0 := avgNPV[1], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgNPV1 := avgNPV[.N], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgDiffElapsed := avgElapsed1 - avgElapsed0]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
asset.ss[, avgDiffNPV := avgNPV1 - avgNPV0]
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
setkey(asset.ss, rd, gid, assignment)
```

```
Error in setkey(asset.ss, rd, gid, assignment): object 'asset.ss' not found
```

```
dim(asset.sss <- asset.ss[!duplicated(asset.ss[, .(rd, gid, assignment)]), ])
```

```
Error in eval(expr, envir, enclos): object 'asset.ss' not found
```

```
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),
                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")))
```

```
Error in ggplot(data = asset.sss, aes(x = avgDiffElapsed, y = avgDiffNPV)): object 'asset.
```

```
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")))
```

```
Error in ggplot(data = asset.sss, aes(x = avgDiffElapsed, y = avgDiffNPV)): object 'asset.
```

VIII 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]
```

```
Error in eval(expr, envir, enclos): object 'tr01' not found
```

```
lstk.ss ← subset(lst, !is.na(gid) & !is.na(elapsed))
```

```
Error in subset(lst, !is.na(gid) & !is.na(elapsed)): object 'lst' not found
```

```
setkey(lstk.ss, rd, gid, assignment)
```

```
Error in setkey(lstk.ss, rd, gid, assignment): object 'lstk.ss' not found
```

```
lstk.ss[, avgElapsed := mean(elapsed, na.rm = T), by = c("rd", "gid", "assignment")]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgElapsed0 := avgElapsed[1], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgElapsed1 := avgElapsed[.N], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgLstkValue := mean(totalLivestockValue, na.rm = T),
        by = c("rd", "gid", "assignment")]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgLstkValue0 := avgLstkValue[1], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgLstkValue1 := avgLstkValue[.N], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[gid == 70204& rd == 1, .(rd, gid, assignment, avgElapsed,
        avgElapsed0, avgElapsed1, avgLstkValue, avgLstkValue0, avgLstkValue1) ]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgDiffElapsed := avgElapsed1 - avgElapsed0]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
lstk.ss[, avgDiffLstkValue := avgLstkValue1 - avgLstkValue0]
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

```
setkey(lstk.ss , rd, gid, assignment)
```

```
Error in setkey(lstk.ss, rd, gid, assignment): object 'lstk.ss' not found
```

```
dim(lstk.sss ← lstk.ss[!duplicated(lstk.ss[, .(rd, gid, assignment)]), ])
```

```
Error in eval(expr, envir, enclos): object 'lstk.ss' not found
```

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")))
```

```
Error in ggplot(data = lstk.sss, aes(x = avgDiffElapsed, y = avgDiffLstkValue)): object 'l
```

```
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),
                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")))
```

```
Error in ggplot(data = lstk.sss, aes(x = avgDiffElapsed, y = avgDiffLstkValue)): object 'l
```

Add assets and livestock.

```
al.ss ← merge(asset.ss, lstk.ss,
        by = c("rd", "gid", "hhid", "assignment", "arm", "elapsed"), all = T)
```

```
Error in merge(asset.ss, lstk.ss, by = c("rd", "gid", "hhid", "assignment", : object 'asse
```

```
al.ss[is.na(assetNPV), assetNPV := 0]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[is.na(totalLivestockValue), totalLivestockValue := 0]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, val := (assetNPV/1000 + totalLivestockValue)]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
setkey(al.ss, rd, gid, assignment)
```

```
Error in setkey(al.ss, rd, gid, assignment): object 'al.ss' not found
```

```
al.ss[, avgElapsed := mean(elapsed, na.rm = T), by = c("rd", "gid", "assignment")]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, avgElapsed0 := avgElapsed[1], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, avgElapsed1 := avgElapsed[.N], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, avgVal := mean(val, na.rm = T), by = c("rd", "gid", "assignment")]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, avgVal0 := avgVal[1], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, avgVal1 := avgVal[.N], by = c("rd", "gid")]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
unique(al.ss[gid == 70101 & rd == 1, .(avgElapsed0, avgElapsed1)])
```

```
Error in unique(al.ss[gid == 70101 & rd == 1, .(avgElapsed0, avgElapsed1)]): object 'al.ss
```

```
al.ss[, avgDiffElapsed := avgElapsed1 - avgElapsed0]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
al.ss[, avgDiffVal := avgVal1 - avgVal0]
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
dim(al.sss ← al.ss[!duplicated(al.ss[, .(rd, gid, assignment)]), ])
```

```
Error in eval(expr, envir, enclos): object 'al.ss' not found
```

```
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",
                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")))
```

```
Error in ggplot(data = al.sss, aes(x = avgDiffElapsed, y = avgDiffVal)): object 'al.sss' n
```

```
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")))
```

```
Error in ggplot(data = al.sss, aes(x = avgDiffElapsed, y = avgDiffVal)): object 'al.sss' n
```

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$",
        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)
```

```
Error in setkey(al.sss, gid, rd): object 'al.sss' not found
```

```
alr.sss ← ro2[al.sss]
```

```
Error in eval(expr, envir, enclos): object 'al.sss' not found
```

```
dim(alr.sss ← alr.sss[!duplicated(alr.sss[, .(rd, gid)]), ])
```

```
Error in eval(expr, envir, enclos): object 'alr.sss' not found
```

```
setkey(alr.sss, gid, rd)
```

```
Error in setkey(alr.sss, gid, rd): object 'alr.sss' not found
```

```
alr.sss[, exist := .N, by = gid]
```

```
Error in eval(expr, envir, enclos): object 'alr.sss' not found
```

```
dim(alr.sss ← alr.sss[exist == 3, ])
```

```
Error in eval(expr, envir, enclos): object 'alr.sss' not found
```

```
destat.alr ← destat(alr.sss[, .(elapsed,
        size, ratioChildren, ratioAdults, ratioDisabled, ratioMale, ratioLiterate,
        ratioLiterateMale, ratioHeadLiterate, avgDiffElapsed,
        avgDiffVal, avgVal, avgVal1, avgVal0, avgElapsed, avgElapsed1, avgElapsed0)])
```

```
Error in destat(alr.sss[, .(elapsed, size, ratioChildren, ratioAdults, : object 'alr.sss'
```

```
destat.alr ← cbind(rownames(destat.alr), destat.alr)
```

```
Error in rownames(destat.alr): object 'destat.alr' not found
```

```
setwd(pathsave)
ltab.alr ← latextab(destat.alr, headercolor = "blue!10",
        alternatecolor = "gray90", delimiterline = NULL,
        hleft = c("\\footnotesize\\hfill", rep("\\footnotesize\\hfil$", ncol(destat.alr)-1
        hcenter = c("2", rep("1.0", ncol(destat.alr)-1)),
        hright = c("", rep("$", ncol(destat.alr)-1)))
```

```
Error in ncol(destat.alr): object 'destat.alr' not found
```

```
write.tablev(ltab.alr, "destat_alr_sss.tex", colnamestrue = F)
```

```
Error in is.data.frame(x): object 'ltab.alr' not found
```

```
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
```

```
Error in cbind(d.rd = 1, alr.sss[rd == 1, .(gid, assignment, arm, elapsed, : object 'alr.s
```

```
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
```

```
Error in cbind(d.rd = 2, alr.sss[rd == 1, .(gid, assignment, arm, elapsed, : object 'alr.s
```

```
dalr.sss ← rbind(dalr.sss1, dalr.sss2)
```

```
Error in rbind(dalr.sss1, dalr.sss2): object 'dalr.sss1' not found
```

```
l1 ← glm(avgDiffVal ~ avgDiffElapsed, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
l2 ← glm(avgDiffVal ~ avgDiffElapsed:arm, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
l3 ← glm(avgDiffVal ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
l4 ← glm(avgDiffVal ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale +
        ratioLiterate + ratioLiterateMale + ratioHeadLiterate, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
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)
```

```
l11 ← glm(avgDiffLstkValue ~ avgDiffElapsed, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
l12 ← glm(avgDiffLstkValue ~ avgDiffElapsed:arm, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
ll3 ← glm(avgDiffLstkValue ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
ll4 ← glm(avgDiffLstkValue ~ avgDiffElapsed:arm +
        size +ratioChildren +ratioAdults +ratioDisabled +ratioMale +
        ratioLiterate + ratioLiterateMale + ratioHeadLiterate, data = dalr.sss)
```

```
Error in is.data.frame(data): object 'dalr.sss' not found
```

```
llinprob ← list(ll1, ll2, ll3, ll4)
```

```
Error in eval(expr, envir, enclos): object 'll1' not found
```

```
llinest ← lapply(llinprob, clx.regobj, Cluster = "gid")
```

```
Error in lapply(llinprob, clx.regobj, Cluster = "gid"): object 'llinprob' not found
```

```
llinest ← lapply(llinest, function(x) x[, -3])
```

```
Error in lapply(llinest, function(x) x[, -3]): object 'llinest' not found
```

```
llinest ← tabs2latex(llinest)
```

```
Error in tabs2latex(llinest): object 'llinest' not found
```

```
R2 ← round(asn(lapply(llinprob,
        function(x) 1-crossprod(summary(x)$deviance.res)/summary(x)$null.dev)), 3)
```

```
Error in lapply(llinprob, function(x) 1 - crossprod(summary(x)$deviance.res)/summary(x)$nu
```

```
en ← asn(lapply(llinprob, function(x) length(x$y)))
```

```
Error in lapply(llinprob, function(x) length(x$y)): object 'llinprob' not found
```

```
rn ← rownames(llinest)
```

```
Error in rownames(llinest): object 'llinest' not found
```

```
rn ← gsub("arm|^se.*", "", rn)
rn ← gsub(":", " * ", rn)
lltab ← rbind(as.matrix(cbind(rn, llinest)), c("$R^{2}$", R2),
        c("$n$", en))
```

```
Error in cbind(rn, llinest): object 'llinest' not found
```

```
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)
```

Table 2: Descriptive statistics of asset regression data

	min	25%	median	75%	max	mean	std	0s	NAs	n
elapsed	49	296	352	556	892	401.6	205.6	0	0	273
size	65	83	89	97	171	92	17	0	0	273
ratioChildren	0.3	0.4	0.4	0.5	0.6	0.4	0.1	0	0	273
ratioAdults	0.4	0.5	0.6	0.6	0.7	0.6	0.1	0	0	273
ratioDisabled	0	0	0	0	0	0	0	165	0	273
ratioMale	0.4	0.5	0.5	0.5	0.6	0.5	0	0	0	273
ratioLiterate	0	0.2	0.3	0.4	0.5	0.3	0.1	0	0	273
ratioLiterateMale	0	0.1	0.2	0.2	0.3	0.2	0.1	0	0	273
ratioHeadLiterate	0	0	0	0	0.1	0	0	51	0	273
avgDiffElapsed	0	339	452.3	543.8	717.4	433.9	155.8	9	0	273
avgDiffVal	−116	−7.1	1	14.1	129.4	2.8	30.4	9	0	273
avgVal	1.2	15.1	42.5	87.1	192.4	56.1	48.1	0	0	273
avgVal1	0.7	15.4	54.8	86.8	230.4	58.9	47.9	0	0	273
avgVal0	1.2	15.1	42.5	87.1	192.4	56.1	48.1	0	0	273
avgElapsed	122.6	266.6	346.2	477.5	892	373.5	146.2	0	0	273
avgElapsed1	544.3	729	835.8	858.2	899	807.4	71.3	0	0	273
avgElapsed0	122.6	266.6	346.2	477.5	892	373.5	146.2	0	0	273

Table 3: DID estimates of asset impacts

rn	(1)	(2)	(3)	(4)
(Intercept)	0.118*	0.143***	0.081	0.164***
	(0.062)	(0.038)	(0.066)	(0.043)
large	−0.037		0.043	
	(0.099)		(0.095)	
large grace	0.093		0.165	
	(0.090)		(0.101)	
cow	−0.079		−0.051	
	(0.101)		(0.101)	
lost to flood	−0.043		−0.005	
	(0.133)		(0.134)	
assignmenttreated		−0.032		−0.047
		(0.024)		(0.079)
assignmentdrop out		−0.204*		−0.224*
		(0.117)		(0.119)
disbursed			0.147**	−0.056
			(0.064)	(0.062)
large * disbursed			−0.366***	
			(0.127)	
large grace * disbursed			−0.228**	
			(0.093)	
cow * disbursed			−0.108	
			(0.134)	
assigncredit				−0.012
				(0.086)
R^2	0.011	0.008	0.023	0.01
n	2024	2024	1800	1800

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.

```
Error in ncol(lltab): object 'lltab' not found
```

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 Statis-*

Table 4: DID estimates of livestock impacts

rn	(1)	(2)	(3)	(4)
(Intercept)	4.499	4.100	−123.340	−101.624
	(5.257)	(5.364)	(242.225)	(232.379)
avgDiffElapsed	−0.008			
	(0.012)			
avgDiffElapsed * traditional		−0.010	−0.024	−0.023
		(0.019)	(0.019)	(0.021)
avgDiffElapsed * large		0.001	− 0.000	−0.001
		(0.014)	(0.014)	(0.015)
avgDiffElapsed * large grace		−0.007	−0.012	−0.009
		(0.015)	(0.016)	(0.017)
avgDiffElapsed * cow		−0.010	−0.009	−0.002
		(0.013)	(0.013)	(0.014)
size			−0.024	0.026
			(0.112)	(0.117)
ratioChildren			54.498	58.120
			(239.679)	(228.340)
ratioAdults			154.896	167.054
			(247.962)	(233.289)
ratioDisabled			−412.698**	−447.624**
			(200.979)	(212.880)
ratioMale			48.160	−35.461
			(42.469)	(55.118)
ratioLiterate				−55.285
				(41.551)
ratioLiterateMale				171.941**
				(84.567)
ratioHeadLiterate				−150.206
				(125.702)
R^2	0.001	0.004	0.039	0.054
n	139	139	139	139

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.

tical 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.