RDRS Bangladesh

Supplemental material regarding:
RDRS and the Poor: Microfinance as Partnership

Aldo Benini
Paul von Bünau
Mozammel Haque
Bhabatosh Nath

September 2011

RDRS Bangladesh
Rangpur Dinajpur Rural Service

RDRS contact details:

RDRS Bangladesh
Rangpur Dinajpur Rural Service
In association with LWF/World Service, Geneva
House 43, Road 10, Sector 6, Uttara
Dhaka-1230, Bangladesh
Telephone +880 2 8954384-86 Fax +880 2 8954391
E-mail: rdrs [at] bangla.net, Website: http://www.rdrsbangla.net

The study was co-published by RDRS and the

RDRS Bangladesh, Jail Road Radhaballav
Rangpur, Bangladesh
Telephone +880 521 62598, 62863

Corresponding author:
Aldo Benini (aldobenini [at] gmail.com)

Contents

List of figures

Purpose

This document details essential data management and statistical operations performed for the analysis of the data used in the RDRS microfinance study. An overview of data extraction from the Microfinance Program database, known as MicroBanker, and of other data (borrower sample survey; Bangladesh consumer price index) as well as of the major types of statistical models is given in the appendix section of the study report. These sections are not reproduced here. However, the extraction script is included in full.

Most of space is taken up with the documentation of statistics calculated in STATA. We reproduce sanitized segments of log files, rather than do-files. This allows us to intersperse selected output. Also, as much as possible, we detail the calculated variables. However, we do not show house-keeping commands (such as *move* or *format*) except where it seems helpful. We do not re-include tables and graphs already given in the study report, but include a few of analytic interest.

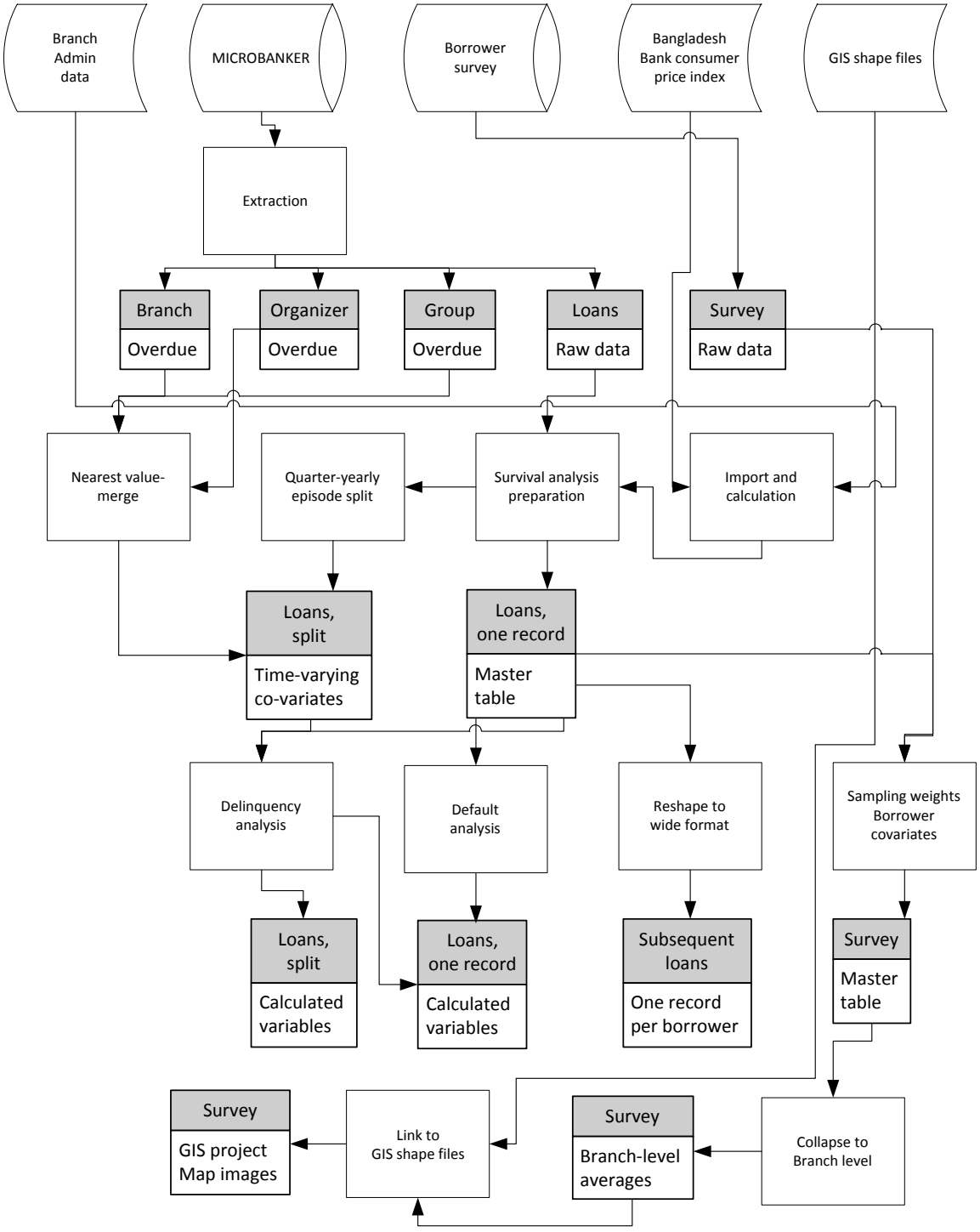
This document thus is a companion to the study report:

Benini, A., P. von Bünau, M. Haque, B. Nath. (2011). RDRS and the Poor: Microfinance as Partnership. Twenty years of microfinance in RDRS Bangladesh. Dhaka and Rangpur, RDRS Bangladesh and North Bengal Institute.

Its major purpose is to make our research reproducible. However, intending users will have to apply to RDRS Bangladesh for permission to obtain and use the data. The address is noted towards the end of this document.

Work flow

Figure 1: Major work flow elements



The diagram shows sources and operations to produce derived data tables and the GIS project. The resulting statistics and maps are not diagrammed here, but are, in large part, detailed in the following sections. Some minor original data tables - unique IDs for branches, organizers, groups and borrowers - are not shown here although, of course, their technical importance was crucial for data management as well as analysis.

MicroBanker data extraction

This SQL script extracts the data from the MicroBanker database, which is an instance of the Microsoft SQL Server 2000 system. Apart from filtering and data conversion, the main computational chore is to compute for each loan, the first date when the borrower has reached a certain level of delinquency. To that end, we need to unravel the whole transaction history and compare it with the installment schedule. In order to compute the monthly on-time-recovery rates (OTR) at group, organizer and branch level, we aggregate the repayment history by months (in the observation period) and to the respective level.

The output consists of the follow four tables.

- LoanDelinquencyHistory: a table of loans; with computed columns that contain the date when the borrower has first fallen delinquent at a certain percentage of the principal

- GroupwiseMonthlyOTR: the monthly on-time-recovery rate for each borrower group (one row for each combination of month in the observation period and borrower group)

- MFOrganizerwiseMonthlyOTR: the monthly on-time-recovery rate for each microfinance organizer (one row for each combination of month in the observation period and borrower group)

- BranchwiseMonthlyOTR: the monthly on-time-recovery rate for each branch (one row for each combination of month in the observation period and branch)

Define stored procedure that computes the last day of a given month

```
CREATE FUNCTION LastDayOfMonth(@DateInMonth AS DATETIME) RETURNS
DATETIME
AS
BEGIN
      DECLARE @CurDate DATETIME;
      SET @CurDate = @DateInMonth;

      WHILE DATEPART(month, @CurDate) = DATEPART(month,@DateInMonth)
      BEGIN
            SET @CurDate = DATEADD(day, 1, @CurDate);
      END

      SET @CurDate = DATEADD(day, -1, @CurDate);
```

```
      RETURN @CurDate;
END
```

Get the name of the branch

```
DECLARE @BranchName NVARCHAR(32);
SET @BranchName = (SELECT BranchName FROM BRParms);
```

Get the current data in the MicroBanker system (the date is tightly controlled by the software)

```
DECLARE @CurRunDate DATETIME;
SET @CurRunDate = (SELECT CurrRunDate FROM BrParms);
```

Get the GIS Id (custom field introduced by RDRS in the branch information table)

```
DECLARE @BranchGISId NVARCHAR(32);
SET @BranchGISId = (SELECT Br_C_Id FROM BRPARMS_RDRS);
```

Get the first available transaction date. This defines the beginning of the observation period. For loans opened before this date, we cannot reconstruct the repayment history.

```
DECLARE @FirstAvailableTransactionDate DATETIME;
SET @FirstAvailableTransactionDate = (SELECT MIN(TrnDate) FROM
TrnHist);
```

Put all transaction (historic and current) together in one table

```
SELECT * INTO #tmpAllTrn FROM
(SELECT recid, Acc, TrnPriAmt, TrnDate, TrnType, BalAmt, TrnAmt
FROM TRNHIST
UNION
SELECT recid, Acc, TrnPriAmt, TrnDate, TrnType, BalAmt, TrnAmt FROM
TRNDAILY) TH
```

Create a table that contains for all loans and all installment dates, the due principal amount and the received amount

List of transaction types

431 Batch Closing
403 Specify Repayment
451 Batch Credit
441 Memo/Voucher Closing
453 Batch Transfer Credit
406 Reverse Closing
409 Memo/Voucher Credit

```
SELECT LA.Acc, LI.DueDate,
```

Due principal amount

```
(SELECT ISNULL(SUM(OrigPriAmt), 0) FROM LnInst WHERE LnInst.Acc=LI.Acc
AND LnInst.DueDate<=LI.DueDate ) AS DuePriAmt,
```

Payments received minus debit adjustments.

```
(SELECT ISNULL(SUM(TrnPriAmt), 0) FROM #tmpAllTrn TR WHERE
TR.Acc=LI.Acc AND TR.TrnDate<=LI.DueDate AND TR.TrnType IN ('403',
'431', '451','441','409','453','406')  )

-(SELECT ISNULL(SUM(TrnPriAmt), 0) FROM #tmpAllTrn TR WHERE
TR.Acc=LI.Acc AND TR.TrnDate<=LI.DueDate AND TR.TrnType IN ('434' ) )
AS PriRecAmt,

LA.GrantedAmtOrig INTO #tmpAccPaidDue
FROM LnInst LI
INNER JOIN LnAcc LA ON LI.Acc=LA.Acc
WHERE LI.DueDate <= @CurRunDate
```

Assemble loan information and compute delinquency dates using the previous table; link loans with group and organizer; decode RDRS-specific categorizations of loans

```
SELECT
  @CurRunDate AS LastObservationDate,
  @BranchName AS BranchName, @BranchGISId AS BranchGISId,
  LA.Acc AS LoanId,
  CASE LA.AccStatus
      WHEN '11' THEN 'Active'
      WHEN '90' THEN 'Matured'
      WHEN '99' THEN 'Closed'
      ELSE LA.AccStatus
  END
  AS LoanStatus,
  CASE CCIF.Type
      WHEN '001' THEN 1
      ELSE 0
  END AS IsIndividualLoan,

  CASE CCIF.TYPE
      WHEN '001' THEN CCIF.Cid
      ELSE (SELECT NULL)
  END AS IndividualCustomerId,
```

Get borrower group id

```
  CASE CCIF.TYPE
      WHEN '001' THEN
        (SELECT Cid FROM RelCid WHERE RelatedCid=CCIF.Cid AND Type IN
('900', 'XXX'))
      ELSE CCif.Cid
  END AS GroupId,
```

Get status of loan

```
  CASE CCIF.TYPE
      WHEN '001' THEN -- Individual loan
        (SELECT CASE C.StatusType
```

```
                  WHEN '001' THEN 'Active'
                  WHEN '999' THEN 'Closed'
                  ELSE 'Dormant'
               END
            FROM RelCid RC
            INNER JOIN CIF C ON RC.RelatedCid=C.CID
            WHERE RC.RelatedCid=CCIF.Cid AND RC.Type IN ('900', 'XXX')
)
      ELSE -- Group loan
        CASE CCIF.StatusType
            WHEN '001' THEN 'Active'
            WHEN '999' THEN 'Closed'
            ELSE 'Dormant'
        END
  END AS GroupStatus,

  CASE CCIF.TYPE
      WHEN '001' THEN -- Individual loan
         CASE CCIF.StatusType
            WHEN '001' THEN 'Active'
            WHEN '999' THEN 'Closed'
            ELSE 'Dormant'
        END
  END AS IndividualCustomerStatus,
```

Get organizer Id

```
  CASE CCIF.TYPE
      WHEN '001' THEN
       (SELECT R2.Cid FROM RelCid R1
            INNER JOIN RelCid R2 ON R2.RelatedCid=R1.CID AND
R2.Type='910'
            WHERE R1.RelatedCid=CCIF.Cid AND R1.Type IN ('900', 'XXX'))

      ELSE
        (SELECT RelatedCid FROM RelCid WHERE CID=CCIF.Cid AND
Type='910')
  END AS MFOrganizerID,
```

Get group gender

```
  CASE CCIF.TYPE
      WHEN '001' THEN
            CASE (SELECT GCIF.CifCode5
                  FROM RelCid RC
                  INNER JOIN CIF GCIF ON GCIF.CID=RC.CID
                  WHERE RC.RelatedCid=CCIF.Cid AND RC.Type IN ('900',
'XXX'))
              WHEN '001' THEN 'Male'
              WHEN '002' THEN 'Female'
              WHEN '003' THEN 'FederationMixed'
              ELSE 'Unknown'
            END
      ELSE
        CASE
            WHEN CCIF.CifCode5 IN ('001') THEN 'Male'
```

```
            WHEN CCIF.CifCode5 IN ('002') THEN 'Female'
            WHEN CCIF.CifCode5 IN ('003') THEN 'FederationMixed'
            ELSE 'Unknown'
        END
  END AS GroupGender,
```

Get category of group (related to loan product/programme)

```
  CASE CCIF.TYPE
      WHEN '001' THEN
            (SELECT UL.FullDesc
                  FROM RelCid RC
                  INNER JOIN CIF GCIF ON GCIF.CID=RC.CID
                  INNER JOIN UserLookup UL ON
GCIF.CifCode1=UL.LookupCode
                  WHERE RC.RelatedCid=CCIF.Cid AND RC.Type IN ('900',
'XXX')
                  AND LookupId='61')
      ELSE
            (SELECT UL.FullDesc
                  FROM UserLookup UL
                  WHERE LookupId='61' AND CCIF.CifCode1=UL.LookupCode)
  END AS GroupCatgory,

  LA.GrantedAmtOrig AS DisbursedAmt,
  LA.OpenDate,
  LA.MatDate AS MaturityDate,
  LA.CumPriPdAmt AS PriAmtPaidAtEndOfObs,
  LA.OduePriAmt AS OduePriAmtAtEndOfObs,
```

Compute date on which the loan was fully repaid

```
 (SELECT MIN(TrnDate) FROM #tmpAllTrn TR
      WHERE BalAmt=0 AND TrnAmt>0 AND TR.Acc=LA.Acc )
            AS LoanRepaidDate,
```

Compute the last cash transaction date (last recovery collection)

```
 (SELECT MAX(TrnDate) FROM #tmpAllTrn TR
      WHERE TrnAmt>0 AND TR.Acc=LA.Acc
          AND TR.TrnType IN ('403', '431', '451') )
            AS LastCashTransactionDate,
```

Compute the balance on the maturity date

```
 CASE WHEN @CurRunDate < LA.MatDate
      THEN (SELECT NULL)
      ELSE
       (
       (SELECT ISNULL(SUM(OrigPriAmt), 0) FROM LnInst WHERE
LnInst.Acc=LA.Acc AND LnInst.DueDate<=LA.MatDate  ) -
       (SELECT ISNULL(SUM(TrnPriAmt), 0) FROM #tmpAllTrn TR WHERE
TR.Acc=LA.Acc AND TR.TrnDate<=LA.MatDate  AND TR.TrnType IN ('403',
'431', '451','441','409','453','406') )
            +(SELECT ISNULL(SUM(TrnPriAmt), 0) FROM #tmpAllTrn TR WHERE
TR.Acc=LA.Acc AND TR.TrnDate<=LA.MatDate  AND TR.TrnType IN ('434' ) )
```

```
      )
    END AS BalanceOnMaturityDate,
```

Compute dates of first delinquency at 1%, 5%, 10%, 20%, 30%, 40%, 50%

```
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.01 ) AS
FirstDateODuePriAmtOver1Pc,
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.05 ) AS
FirstDateODuePriAmtOver5Pc,
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.10 ) AS
FirstDateODuePriAmtOver10Pc,
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.20 ) AS
FirstDateODuePriAmtOver20Pc,
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.30 ) AS
FirstDateODuePriAmtOver30Pc,
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.40 ) AS
FirstDateODuePriAmtOver40Pc,
 (SELECT MIN(DueDate) FROM #tmpAccPaidDue PD WHERE PD.Acc=LA.Acc
   AND (PD.DuePriAmt-PD.PriRecAmt)/PD.GrantedAmtOrig  >= 0.50 ) AS
FirstDateODuePriAmtOver50Pc
INTO LoanDelinquencyHistory
FROM LNAcc LA
INNER JOIN RelAcc RA ON LA.Acc=RA.Acc
INNER JOIN CIF CCIF ON RA.CID=CCIF.CID
WHERE LA.OpenDate >= @FirstAvailableTransactionDate
ORDER BY LA.OpenDate
```

Compute a table that contains for each month in the observation period the first and last day; this table is needed to compute the OTRs.

```
CREATE TABLE #tmpPeriods ( PeriodStartDate DATETIME, PeriodEndDate
DATETIME );

INSERT INTO #tmpPeriods VALUES (@FirstAvailableTransactionDate,
      dbo.LastDayOfMonth(@FirstAvailableTransactionDate));

WHILE ( SELECT MAX(PeriodStartDate) FROM #tmpPeriods ) < @CurRunDate

BEGIN
      DECLARE @StartNewPeriod DATETIME;
      SET @StartNewPeriod = DATEADD(month, 1, (SELECT
MAX(PeriodStartDate) FROM #tmpPeriods));
      INSERT INTO #tmpPeriods VALUES (@StartNewPeriod,
dbo.LastDayOfMonth(@StartNewPeriod));
END
DELETE FROM #tmpPeriods WHERE PeriodEndDate >= @CurRunDate;
```

Create a loanwise monthly recovery table, which is then aggregated

```
SELECT @BranchName AS BranchName, @BranchGISId AS BranchGISId,
DisbursedAmt, P.PeriodStartDate, P.PeriodEndDate, L.LoanId,
L.LoanStatus, L.MFOrganizerId, L.GroupId,
```

Payment received in month

```
  ( SELECT ISNULL(SUM(TH.TrnPriAmt), 0) FROM #tmpAllTrn TH
      WHERE TH.Acc=L.LoanID
      AND TH.TrnDate>=P.PeriodStartDate AND TH.TrnDate<=P.PeriodEndDate
AND
      TH.TrnType IN ('403', '431', '451', '441', '409', '453', '406') )
```

Minus debit in month

```
-( SELECT ISNULL(SUM(TH.TrnPriAmt), 0)
      FROM #tmpAllTrn TH
      WHERE TH.Acc=L.LoanID
      AND TH.TrnDate>=P.PeriodStartDate AND TH.TrnDate<=P.PeriodEndDate
AND
      TH.TrnType IN ('434') )
      AS PaymentRecvInPeriod,
```

Payments received in all previous months

```
  ( SELECT ISNULL(SUM(TH.TrnPriAmt), 0)
      FROM #tmpAllTrn TH
      WHERE TH.Acc=L.LoanId
      AND TH.TrnDate < P.PeriodStartDate AND
      TH.TrnType IN ('403', '431', '451', '441','409','453','406') )
```

Minus debit in previous months

```
  -( SELECT ISNULL(SUM(TH.TrnPriAmt), 0)
      FROM #tmpAllTrn TH
      WHERE TH.Acc=L.LoanID
      AND TH.TrnDate < P.PeriodStartDate AND
      TH.TrnType IN ('434') )
```

Minus all due principal from previous months

```
  -( SELECT ISNULL(SUM(LI.OrigPriAmt), 0)
        FROM LnInst LI
      WHERE LI.Acc=L.LoanId
      AND LI.DueDate < P.PeriodStartDate ) AS
BalancePriPaymentsVsDueAtStartOfPeriod,
```

New due principal in this month

```
  ( SELECT ISNULL(SUM(LI.OrigPriAmt), 0)
        FROM LnInst LI
      WHERE LI.Acc=L.LoanId
      AND LI.DueDate >= P.PeriodStartDate AND LI.DueDate <=
P.PeriodEndDate ) AS PriDueInThisPeriod
INTO LoanwiseMonthlyRecovery
FROM LoanDelinquencyHistory L, #tmpPeriods P
```

Calculate advance at the start of the month

```
ALTER TABLE LoanwiseMonthlyRecovery ADD AdvanceAtStartOfPeriod NUMERIC;
GO

UPDATE LoanwiseMonthlyRecovery SET AdvanceAtStartOfPeriod
      = CASE WHEN BalancePriPaymentsVsDueAtStartOfPeriod > 0 THEN
                  BalancePriPaymentsVsDueAtStartOfPeriod
            ELSE 0
      END
```

Compute overdue.

```
ALTER TABLE LoanWiseMonthlyRecovery ADD OverdueAmount NUMERIC;
GO

UPDATE LoanWiseMonthlyRecovery SET OverdueAmount
  = CASE WHEN
            (BalancePriPaymentsVsDueAtStartOfPeriod -
PriDueInThisPeriod + PaymentRecvInPeriod) < 0
            THEN
            -(BalancePriPaymentsVsDueAtStartOfPeriod -
PriDueInThisPeriod + PaymentRecvInPeriod)
            ELSE 0
            END;
```

Calculate monthly recovery: this is the sum of the advance and the cash received in this period; capped to the amount of the due principal.

```
ALTER TABLE LoanwiseMonthlyRecovery ADD TotalRecoveryInPeriod NUMERIC;
GO

UPDATE LoanwiseMonthlyRecovery SET TotalRecoveryInPeriod
 = CASE WHEN AdvanceAtStartOfPeriod+PaymentRecvInPeriod
            <= PriDueInThisPeriod
      THEN AdvanceAtStartOfPeriod+PaymentRecvInPeriod
             ELSE PriDueInThisPeriod
      END;
```

Create Branchwise monthly OTR table.

```
SELECT BranchName, BranchGISId, PeriodStartDate, PeriodEndDate,
      SUM(TotalRecoveryInPeriod)/SUM(PriDueInThisPeriod) AS
OnTimeRecoveryRate,
      SUM(OverdueAmount)/SUM(DisbursedAmt) AS OverduePercOfPrincipal
INTO BranchwiseMonthlyOTR
FROM LoanwiseMonthlyRecovery
GROUP BY BranchName, PeriodStartDate, PeriodEndDate, BranchGISId
HAVING SUM(PriDueInThisPeriod)>0
ORDER BY PeriodStartDate;
```

Create MFOrganizer-wise monthly OTR table.

```
SELECT BranchName, BranchGISId AS BranchGISId, MFOrganizerID,
PeriodStartDate, PeriodEndDate,
      SUM(TotalRecoveryInPeriod)/SUM(PriDueInThisPeriod) AS
OnTimeRecoveryRate,
      SUM(OverdueAmount)/SUM(DisbursedAmt) AS OverduePercOfPrincipal
INTO MFOrganizerwiseMonthlyOTR
FROM LoanwiseMonthlyRecovery
GROUP BY BranchName, PeriodStartDate, PeriodEndDate, MFOrganizerID,
BranchGISId
HAVING SUM(PriDueInThisPeriod)>0
ORDER BY PeriodStartDate;
```

Create Group-wise monthly OTR table.

```
SELECT BranchName, BranchGISId AS BranchGISId, GroupId,
PeriodStartDate, PeriodEndDate,
      SUM(TotalRecoveryInPeriod)/SUM(PriDueInThisPeriod) AS
OnTimeRecoveryRate,
      SUM(OverdueAmount)/SUM(DisbursedAmt) AS OverduePercOfPrincipal
INTO GroupwiseMonthlyOTR
FROM LoanwiseMonthlyRecovery
GROUP BY BranchName, PeriodStartDate, PeriodEndDate, GroupId,
BranchGISId
HAVING SUM(PriDueInThisPeriod)>0
ORDER BY PeriodStartDate;
```

Source files

Loan data

The original data extracted and computed from MicroBanker data arrived as C:\...\Data\MF\110524_Paul\MB_loan_delinquency_data_May23.mdb

The tables of the Access database were converted into STATA tables.

C:\...\Data\MF\110524_Aldo_3formats\Branches.dta
C:\...\Data\MF\110524_Aldo_3formats\BranchwiseMonthlyOTR.dta

C:\...\Data\MF\110524_Aldo_3formats\MFOrganizers.dta
C:\...\Data\MF\110524_Aldo_3formats\MFOrganizerwiseMonthlyOTR.dta

C:\...\Data\MF\110524_Aldo_3formats\Groups.dta
C:\...\Data\MF\110524_Aldo_3formats\GroupwiseMonthlyOTR.dta

C:\...\Data\MF\110524_Aldo_3formats\IndividualCustomers.dta
C:\...\Data\MF\110524_Aldo_3formats\LoanDelinquencyHistory.dta

Branch administrative data and branch area polygon files

C:\...\Data\MF\110526_BranchCodes\110526_1430AB_BranchAndAdminNamesUpdated.dta
C:\...\RDRS_MF2011\ rdrsmfbranches.shp and affiliated files.

Consumer price index

C:\...\Data\CPI\CPI_BoB_uptoAug2010.pdf

Borrower survey

C:\...\Data\MF\110616_Paul_BorrowerSurvey\ BorrowerSurvey_0615_1500.mdb

Master files

We keep four master files:

Loans

C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep2010_tagsCORR_work07.dta

Quarter-yearly episode-split loans

C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1549AB_IndivLoans_10pcDelinq_EpisodeSplit_work06.dta

Branch level averages for GIS purposes

C:\...\Analyses\Analyses110618_3_CollapseToBranches\110618_1249AB_BranchLevelSummary_work11.dta

Borrower sample survey

C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1603AB_BorrowerSurvey_and_LoanInfo_work10.dta

Loan analysis

Global observation period

The final loan master table includes borrowers and their loans if the first loan was disbursed on or after June 13, 2004. This date was copied from the opening date of a preceding small study using data only from the Thakurgaon Unit. The exclusions due to this cut-off point were few. We observe borrowers and loans until September 30, 2010, a very small number until October 2, 2010. Observations from five branches were excluded because the variable "overdue at the end of the observation period" was calculated for dates several months later.

Pre-processing loan data

```
 [From                                  log                                        file:
C:\...\Analyses\Analyses110524BasicLoanFile\110524_2052AB_IndividLoanTable_to_stset.log]


. *-------------------------------------------------------------------------------
. * RDRS MICROBANKER LOAN-WISE DATA UP TO SEPTEMBER 30, 2010
.
. * PRE-PROCESS RAW LOAN DATA FILE UP TO MULTIPLE-FAILURE STSET POINT
. * for all delinquency levels for which "first delinquency dates" were computed
.
. * This table excludes two categories of loans:
. * 1. Loans from seven branches with last observation date after 2 October 2010
. * 2. Group loans
```

```
. *-------------------------------------------------------------------------------
```

Data source

```
. * DATA SOURCE AND WORKING FILE NAME AND DESTINATION
.
. * All-RDRS data, converted from Access file, was stored as:

"C:\...\Data\MF\110524_Aldo_3formats\LoanDelinquencyHistory.dta

.
. * Working copy:
. save "C:\...\Analyses\Analyses110524BasicLoanFile\110524_2037AB_IndivLoans_work02.dta",
replace

. *-------------------------------------------------------------------------------
.
. count
1143146

. * Number of loans before excluding those two groups.
.
. *-------------------------------------------------------------------------------
. * PRE-PROCESSING
.
. * Create record identifier to preserve initial sort order:
. gen recno = _n
. . .
size:   225,199,762
[List of original variables. Will be reordered and labeled, see further below]
--------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------
--------------------
recno           float  %9.0g                 Record identifier
LastObservati~e long   %tdD_m_Y
BranchName      str21  %21s
BranchGISId     str3   %3s
LoanId          str7   %7s
LoanStatus      str7   %7s
IsIndividualL~n long   %12.0g
IndividualC~rId long   %12.0g
GroupId         str6   %6s
GroupStatus     str7   %7s
IndividualCus~s str7   %7s
MFOrganizerID   str6   %6s
GroupGender     str15  %15s
GroupCatgory    str18  %18s
DisbursedAmt    float  %9.0g
OpenDate        long   %tdD_m_Y
MaturityDate    long   %tdD_m_Y
PriAmtPaidAtE~s float  %9.0g
OduePriAmtAtE~s float  %9.0g
LoanRepaidDate  long   %tdD_m_Y
LastCashTrans~e long   %tdD_m_Y
BalanceOnMatu~e float  %9.0g
FirstDateOD~1Pc long   %tdD_m_Y
FirstDateOD~5Pc long   %tdD_m_Y
FirstDateO~10Pc long   %tdD_m_Y
FirstDateO~20Pc long   %tdD_m_Y
FirstDateO~30Pc long   %tdD_m_Y
FirstDateO~40Pc long   %tdD_m_Y
FirstDateO~50Pc long   %tdD_m_Y
LoanGId         long   %12.0g
BranchGId       long   %12.0g
MFOrganizerGid  long   %12.0g
IndividualC~GId long   %12.0g
GroupGId        long   %12.0g
--------------------
. [House-keeping not shown]
```

Core sample flag

```
. * CREATION OF A VARIABLE "INCLUDE" FOR THE LATER CORE SAMPLE
.
. * At this stage, we include: All records that have Organizer ID and a group ID

. * Variable for inclusion in analysis:
```

```
. gen byte include = ( MFOrganizerGid ~= . &  GroupGId ~= .)

. label var include "Included if MFOrganizerGid and GroupGId both present"

. tab include

Included if |
MFOrganizer |
    Gid and |
   GroupGId |
       both |
    present |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    205,803       18.00       18.00
          1 |    937,343       82.00      100.00
------------+-----------------------------------
      Total |  1,143,146      100.00
```

Exclusions

Group loans

```
. * Group loans:
. tab IsIndividualLoan

IsIndividua |
      lLoan |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    138,733       12.14       12.14
          1 |  1,004,413       87.86      100.00
------------+-----------------------------------
      Total |  1,143,146      100.00

. drop if IsIndividualLoan == 0
(138733 observations deleted)

. count
1004413

. count if include
937343
```

Branches with defective last observation date

```
. * Loans by branches with dates of last observation after 2 October 2010:
. tab LastObservationDate

    Date of |
       last |
observation |      Freq.     Percent        Cum.
------------+-----------------------------------
  01 Sep 10 |      2,832        0.28        0.28
  30 Sep 10 |    937,979       93.39       93.67
  02 Oct 10 |      8,538        0.85       94.52
  30 Dec 10 |     35,068        3.49       98.01
  22 Feb 11 |     12,123        1.21       99.22
  30 Apr 11 |      7,873        0.78      100.00
------------+-----------------------------------
      Total |  1,004,413      100.00

. drop if LastObservationDate > mdy(10,02,2010)
(55064 observations deleted)

. count
949349

. count if include
884126

. * [Note: A later file will exclude also those without Organizer or group ID]
```

Delinquency-based failure event definitions; *stset*

End of observation period

```
. * FAILURE DEFINITION, END OF OBSERVATION PERIOD, STSET, STDES, STSUM FOR EACH
DELINQUENCY LEVEL
. *-------------------------------------------------------------------------
. * 1. Common elements:
.
. * End of observation period:
. * Considerable time may elapse in the RDRS MF admin between date of last cash
transaction and formal repayment date

. gen  DiffLastTransVsRepaidDate =  LoanRepaidDate - LastCashTransactionDate
. . .

. summ DiffLastTransVsRepaidDate, detail

      Delay (in days) betw. last cash transaction and
                    formal replayment
-------------------------------------------------------------
      Percentiles      Smallest
 1%             0         -1624
 5%             0         -1451
10%             0         -1371       Obs              631701
25%             0         -1308       Sum of Wgt.      631701

50%             0                     Mean           25.00385
                        Largest       Std. Dev.      114.3221
75%             0          2000
90%            23          2008       Variance       13069.53
95%           122          2013       Skewness       6.130481
99%           669          2032       Kurtosis       48.61848

.
. * Thus, the logic of defining the end date of the observation period is:
. ** If delinquent, then = date first delinquent.
. ** If not delinquent and repaid, then = minimum(date last cash transaction, date formal
repayment).
. ** If not delinquent and not repaid, then = end of global observation period, which we
set for all = mdy(9,30,2010)
.
. * Sort order critical:
. gsort  IndividualCustomerGId OpenDate

.
. * Label for failure variables:
. label define failure 0 "Loan running non-delinquent" 1 "Loan repaid non-delinquent"  2
"Loan delinquent"
```

stset demonstrated with the 1-percent delinquency model

Note on system variables created by stset

The definition of survival analysis models using the *stset* command creates these five system variables:

```
. label var  _t0 "Time from origin observation starts"

. label var _t "Time from origin event occurred, or end obs period"

. label var _origin "Calender date first observation starts"

. label var _d "Failure event occurred"

. label var _st "Included in survival model calculation"
.
```

However, these are re-created from scratch every time the data is *stset*-ed differently, which means also that they shed their labels.

Demonstration

```
. *-------------------------------------------------------------------------
```

```
. * 2. Delinquency level-specific stset, etc. and some auxiliary variables:
. *-------------------------------------------------------------------------
.
[documented here for the 1 percent delinquency level; other definitions analogous but not
shown here for space reasons:]

. * For 01pc delinquency:
.
. * Failure definition:
.
. gen byte failure01pc = 2

. * 2 is delinquency = failure event.
.
. replace  failure01pc = 1 if FirstDateODuePriAmtOver01Pc == . &  LoanRepaidDate ~= .

. * 1 loan was repaid, without delinquency, during observation period.
.
. replace  failure01pc = 0 if FirstDateODuePriAmtOver01Pc == . &  LoanRepaidDate == .

. * 0 loan running at end of observation, no delinquency
.
. label var  failure01pc "Failure event (01pc delinquent)"

. label val  failure01pc failure

. tab failure01pc, missing

       Failure event (01pc |
               delinquent) |      Freq.     Percent        Cum.
---------------------------+-----------------------------------
Loan running non-delinquent|    149,145       15.71       15.71
Loan repaid non-delinquent |    268,057       28.24       43.95
          Loan delinquent |    532,147       56.05      100.00
---------------------------+-----------------------------------
                    Total |    949,349      100.00

.
. * Define end of observation period:
. gen Delinq01pcOrEnddate  = FirstDateODuePriAmtOver01Pc

.  replace  Delinq01pcOrEnddate    =  min(  LoanRepaidDate,  LastCashTransactionDate)  if
FirstDateODuePriAmtOver01Pc == . &  LoanRepaidDate ~= .

. replace Delinq01pcOrEnddate  =  mdy(9,30,2010)  if   FirstDateODuePriAmtOver01Pc ==. &
LoanRepaidDate ==.

. format %tdD_m_Y Delinq01pcOrEnddate

. label var Delinq01pcOrEnddate "End of observation period (01pc delinq model)"

. codebook  Delinq01pcOrEnddate

-------------------------------------------------------------------------------------------
Delinq01pcOrEnddate                         End of observation period (01pc delinq model)
-------------------------------------------------------------------------------------------

                  type:  numeric daily date (float)

                 range:  [15362,18535]                units:  1
       or equivalently:  [22jan2002,30sep2010]        units:  days
         unique values:  2298                     missing .:  0/949349
. . .
```

stset, stdes, stsum

```
. stset  Delinq01pcOrEnddate if  include, id(IndividualCustomerGId)
failure(failure01pc==2) time0(OpenDate) exit(time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure01pc == 2
obs. time interval:  (OpenDate, Delinq01pcOrEnddate]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate
            if exp:  include

--------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
```

```
      526  entry on or after exit (OpenDate>Delinq01pcOrEnddate)PROBABLE ERROR
    56542  overlapping records (Delinq01pcOrEnddate[_n-1]>OpenDate)PROBABLE ERROR
--------------------------------------------------------------------------------
   827058  obs. remaining, representing
   360690  subjects
   466649  failures in multiple failure-per-subject data
 1.53e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      3199

. stdes

        failure _d:  failure01pc == 2
  analysis time _t:  (Delinq01pcOrEnddate-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

                                |-------------- per subject --------------|
Category                   total        mean         min     median        max
--------------------------------------------------------------------------------
no. of subjects           360690
no. of records            827058    2.292988           1          2         14

(first) entry time                  8.004616           0          0       1612
(final) exit time                   636.7759           1        492       3199

subjects with gap         219107
time on gap if gap       74069887    160.5291           1        119       2090
time at risk           1.527e+08    423.4151           1        333       2261

failures                  466649    1.293768           0          1          9
--------------------------------------------------------------------------------

. stsum

        failure _d:  failure01pc == 2
  analysis time _t:  (Delinq01pcOrEnddate-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

        |               incidence       no. of    |------ Survival time -----|
        | time at risk     rate        subjects        25%       50%       75%
--------+-----------------------------------------------------------------------
   total |    152721610   .0030556        360690         92       294       473

. * Keep set available for later comparisons or sub-setting:
. clonevar stset01pcdel = _st

. label var stset01pcdel "Loan sample of the 01pc delinquency model"

. move stset01pcdel LoanGId
```

Adjust end dates to minimize overlapping records

```
. * Overlapping records were excluded. To reduce information loss,
. * Generate dummy for overlapping with next record [this again is model-specific; here
we are dealing with 01pc delinqency]:

.     gen     byte     OverlapsWithNext01pcDel     =         (IndividualCustomerGId     ==
IndividualCustomerGId[_n+1] & Delinq01pcOrEnddate >  OpenDate[_n+1])

. label var OverlapsWithNext01pcDel "Obs period overlaps with next record (01pc delinq
model)"

. tab  OverlapsWithNext01pcDel if include

 Obs period |
   overlaps |
  with next |
     record |
      (01pc |
     delinq |
     model) |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    828,053       93.66       93.66
          1 |     56,073        6.34      100.00
------------+-----------------------------------
      Total |    884,126      100.00
```

```
.
. gen byte WasDelinq01pc = (FirstDateODuePriAmtOver01Pc ~= .)

. label var WasDelinq01pc "Loan was delinquent > 01pc of principal"


. tab  WasDelinq01pc if include

   Loan was |
 delinquent |
  > 01pc of |
  principal |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    398,984       45.13       45.13
          1 |    485,142       54.87      100.00
------------+-----------------------------------
      Total |    884,126      100.00

.
. tab  OverlapsWithNext01pcDel WasDelinq01pc if include, V

Obs period |
  overlaps |
 with next |
    record |
     (01pc | Loan was delinquent >
    delinq |   01pc of principal
    model) |         0          1 |     Total
-----------+----------------------+----------
         0 |   364,659    463,394 |   828,053
         1 |    34,325     21,748 |    56,073
-----------+----------------------+----------
     Total |   398,984    485,142 |   884,126

               Cramér's V =  -0.0841

.
. * Create adjusted end of observation period such that =< OpenDate[_n+1], in order to
minimize record exclusions:

. clonevar  Delinq01pcOrEnddateAdj =  Delinq01pcOrEnddate

. label var Delinq01pcOrEnddateAdj "End of observation period (01pc - adj for overlap)"

. replace  Delinq01pcOrEnddateAdj =  OpenDate[_n+1] if  OverlapsWithNext01pcDel
```

Adjusted stset, stdes, stsum

```
.    stset       Delinq01pcOrEnddateAdj    if       include,    id(IndividualCustomerGId)
failure(failure01pc==2) time0(OpenDate
> ) exit(time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure01pc == 2
obs. time interval:  (OpenDate, Delinq01pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate
            if exp:  include

--------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2650  entry on or after exit (OpenDate>Delinq01pcOrEnddateAdj)PROBABLE ERROR
--------------------------------------------------------------------------------
   881476  obs. remaining, representing
   360690  subjects
   483722  failures in multiple failure-per-subject data
 1.63e+08  total analysis time at risk, at risk from t =         0
                              earliest observed entry t =         0
                                   last observed exit t =      3199

. stdes

         failure _d:  failure01pc == 2
   analysis time _t:  (Delinq01pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                id:  IndividualCustomerGId
```

```
                                |-------------- per subject --------------|
Category                  total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects          360690
no. of records           881476     2.44386           1          2         15

(first) entry time                .0616568           0          0        476
(final) exit time                  642.936           1        504       3199

subjects with gap        219003
time on gap if gap      68496694    148.4216           1         95       2090
time at risk           1.634e+08    452.9697           1        349       2272

failures                 483722    1.341102           0          1         10
------------------------------------------------------------------------------

. stsum

        failure _d:  failure01pc == 2
  analysis time _t:  (Delinq01pcOrEnddateAdj-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

        |              incidence       no. of    |------ Survival time -----|
        | time at risk     rate        subjects        25%       50%       75%
--------+---------------------------------------------------------------------
   total |    163381643   .0029607        360690         97       296       490

. clonevar stset01pcdelAdj = _st

. label var stset01pcdelAdj "Loan sample of the 01pc model (adj. for overlap)"
```

Comparison of unadjusted and adjusted end of observation dates

```
-------------------
Delinq01pcOrEnddate                       End of observation period (01pc delinq model)
-------------------

                  type:  numeric daily date (float)

                 range:  [15362,18535]                units:  1
       or equivalently:  [22jan2002,30sep2010]        units:  days
         unique values:  2298                     missing .:  0/949349

                  mean:   17807.8 = 02oct2008 (+ 18 hours)
             std. dev:   587.888

          percentiles:        10%       25%       50%       75%       90%
                           16969     17342     17898     18357     18535
                         17jun2006 25jun2007 01jan2009 05apr2010 30sep2010

-------------------
Delinq01pcOrEnddateAdj                  End of observation period (01pc - adj for overlap)
-------------------

                  type:  numeric daily date (float)

                 range:  [15362,18535]                units:  1
       or equivalently:  [22jan2002,30sep2010]        units:  days
         unique values:  2300                     missing .:  0/949349

                  mean:   17793.2 = 18sep2008 (+ 5 hours)
             std. dev:   583.965

          percentiles:        10%       25%       50%       75%       90%
                           16960     17331     17881     18323     18535
                         08jun2006 14jun2007 15dec2008 02mar2010 30sep2010

. *------------------------------------------------------------------------------
. * [The other models work similarly. We show only the adjusted end of observation
period-related stset, stdes and stsum results:]
```

Fewer details for the 5, 10, 20, 30, 40 and 50 percent delinquency models

stset, stdes, stsum - only with the *adjusted* end of observation period date

05pc delinquency

```
.    stset        Delinq05pcOrEnddateAdj    if        include,    id(IndividualCustomerGId)
failure(failure05pc==2) time0(OpenDate) exit(time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure05pc == 2
obs. time interval:  (OpenDate, Delinq05pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate
            if exp:  include

------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2696  entry on or after exit (OpenDate>Delinq05pcOrEnddateAdj)PROBABLE ERROR
------------------------------------------------------------------------------
   881430  obs. remaining, representing
   360686  subjects
   304047  failures in multiple failure-per-subject data
 2.10e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      3199

. stdes

         failure _d:  failure05pc == 2
   analysis time _t:  (Delinq05pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                id:  IndividualCustomerGId

                                  |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           360686
no. of records            881430     2.44376           1          2         15

(first) entry time                    .06995           0          0        476
(final) exit time                    689.8437          1        567       3199

subjects with gap         212376
time on gap if gap      38424403    89.29532           1         15       2034
time at risk           2.104e+08    583.2423           1        481       2301

failures                  304047    .8429687           0          1         10
------------------------------------------------------------------------------

. stsum

         failure _d:  failure05pc == 2
   analysis time _t:  (Delinq05pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                id:  IndividualCustomerGId

         |               incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
   total |    210367325   .0014453        360686        279       530       921

. *------------------------------------------------------------------------------
.
```

10pc delinquency

```
.    stset        Delinq10pcOrEnddateAdj    if        include,    id(IndividualCustomerGId)
failure(failure10pc==2) time0(OpenDate) exit(time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure10pc == 2
obs. time interval:  (OpenDate, Delinq10pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate
```

```
            if exp:  include

------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2669  entry on or after exit (OpenDate>Delinq10pcOrEnddateAdj)PROBABLE ERROR
------------------------------------------------------------------------------
   881457  obs. remaining, representing
   360687  subjects
   170642  failures in multiple failure-per-subject data
 2.41e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      3199

. stdes

         failure _d:  failure10pc == 2
   analysis time _t:  (Delinq10pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

                                |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           360687
no. of records            881457    2.443828           1          2         15

(first) entry time                  .0570439           0          0        372
(final) exit time                   731.6454           1        608       3199

subjects with gap         206536
time on gap if gap      22711597    56.33871           1         11       2006
time at risk           2.412e+08    668.6207           1        573       2301

failures                  170642    .4731027           0          0          9
------------------------------------------------------------------------------

. stsum

         failure _d:  failure10pc == 2
   analysis time _t:  (Delinq10pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

         |               incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
   total |    241162809   .0007076        360687        471       928      1735

. *------------------------------------------------------------------------------
.
```

20pc delinquency

```
.    stset        Delinq20pcOrEnddateAdj    if        include,    id(IndividualCustomerGId)
failure(failure20pc==2) time0(OpenDate
> ) exit(time .) origin(time OpenDate)

                 id:  IndividualCustomerGId
       failure event:  failure20pc == 2
obs. time interval:  (OpenDate, Delinq20pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
             origin:  time OpenDate
             if exp:  include

------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2693  entry on or after exit (OpenDate>Delinq20pcOrEnddateAdj)PROBABLE ERROR
------------------------------------------------------------------------------
   881433  obs. remaining, representing
   360682  subjects
    89779  failures in multiple failure-per-subject data
 2.59e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      3199

. stdes
```

```
        failure _d:  failure20pc == 2
  analysis time _t:  (Delinq20pcOrEnddateAdj-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

                                  |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           360682
no. of records            881433    2.443795           1          2         15

(first) entry time                  .0599808           0          0        399
(final) exit time                   768.2947           1        641       3199

subjects with gap         203779
time on gap if gap      17764554    45.32815           1          9       1971
time at risk           2.593e+08    718.9821           1        614       2440

failures                   89779    .2489146           0          0          8
------------------------------------------------------------------------------

. stsum

        failure _d:  failure20pc == 2
  analysis time _t:  (Delinq20pcOrEnddateAdj-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

        |               incidence       no. of    |------ Survival time -----|
        | time at risk     rate        subjects        25%       50%       75%
--------+---------------------------------------------------------------------
   total |    259323884   .0003462        360682        803      1855         .

. *------------------------------------------------------------------------------
```

30pc delinquency

```
.    stset       Delinq30pcOrEnddateAdj    if       include,    id(IndividualCustomerGId)
failure(failure30pc==2) time0(OpenDate
> ) exit(time .) origin(time OpenDate)

               id:  IndividualCustomerGId
    failure event:  failure30pc == 2
obs. time interval:  (OpenDate, Delinq30pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate
            if exp:  include

------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2700  entry on or after exit (OpenDate>Delinq30pcOrEnddateAdj)PROBABLE ERROR
------------------------------------------------------------------------------
   881426  obs. remaining, representing
   360681  subjects
    55652  failures in multiple failure-per-subject data
 2.71e+08  total analysis time at risk, at risk from t =         0
                            earliest observed entry t =         0
                                 last observed exit t =      3199

. stdes

        failure _d:  failure30pc == 2
  analysis time _t:  (Delinq30pcOrEnddateAdj-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

                                  |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           360681
no. of records            881426    2.443783           1          2         15

(first) entry time                  .0618386           0          0        670
(final) exit time                   795.4804           1        665       3199
```

```
subjects with gap         202741
time on gap if gap      16054944    41.39035           1          9       1943
time at risk           2.708e+08    750.9057           1        638       3163

failures                   55652     .154297           0          0          6
------------------------------------------------------------------------------

. stsum

         failure _d:  failure30pc == 2
   analysis time _t:  (Delinq30pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

         |              incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
   total |    270837418   .0002055        360681       1283         .         .

. *-----------------------------------------------------------------------------
```

40pc delinquency

```
.    stset        Delinq40pcOrEnddateAdj    if        include,    id(IndividualCustomerGId)
failure(failure40pc==2) time0(OpenDate
> ) exit(time .) origin(time OpenDate)

                 id:  IndividualCustomerGId
      failure event:  failure40pc == 2
obs. time interval:  (OpenDate, Delinq40pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
             origin:  time OpenDate
             if exp:  include

------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2707  entry on or after exit (OpenDate>Delinq40pcOrEnddateAdj)PROBABLE ERROR
------------------------------------------------------------------------------
   881419  obs. remaining, representing
   360681  subjects
    37054  failures in multiple failure-per-subject data
 2.80e+08  total analysis time at risk, at risk from t =         0
                              earliest observed entry t =         0
                                   last observed exit t =      3199

. stdes

         failure _d:  failure40pc == 2
   analysis time _t:  (Delinq40pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

                                  |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           360681
no. of records            881419    2.443763           1          2         15

(first) entry time                   .0618386           0          0        670
(final) exit time                    817.9657           1        682       3199

subjects with gap         202083
time on gap if gap      15107735    39.16925           1          9       1878
time at risk           2.799e+08    776.0171           1        655       3163

failures                   37054    .1027334           0          0          6
------------------------------------------------------------------------------

. stsum

         failure _d:  failure40pc == 2
   analysis time _t:  (Delinq40pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

         |              incidence       no. of    |------ Survival time -----|
```

```
         | time at risk     rate        subjects        25%       50%       75%
---------+-------------------------------------------------------------------
   total |    279894631   .0001324        360681       2191         .         .

. *-------------------------------------------------------------------------------
```

50pc delinquency

```
.
.    stset       Delinq50pcOrEnddateAdj    if       include,    id(IndividualCustomerGId)
failure(failure50pc==2) time0(OpenDate
> ) exit(time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure50pc == 2
obs. time interval:  (OpenDate, Delinq50pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate
            if exp:  include

--------------------------------------------------------------------------------
   949349  total obs.
    65223  ignored at outset because of -if <exp>-
     2708  entry on or after exit (OpenDate>Delinq50pcOrEnddateAdj)PROBABLE ERROR
--------------------------------------------------------------------------------
   881418  obs. remaining, representing
   360681  subjects
    24834  failures in multiple failure-per-subject data
 2.87e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      3199

. stdes

         failure _d:  failure50pc == 2
   analysis time _t:  (Delinq50pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

                                   |-------------- per subject --------------|
Category                    total        mean         min     median         max
--------------------------------------------------------------------------------
no. of subjects            360681
no. of records             881418    2.443761           1          2          15

(first) entry time                    .0629171           0          0         670
(final) exit time                      836.614           1        701        3199

subjects with gap          201679
time on gap if gap       14481404    37.67138           1          8        1817
time at risk            2.872e+08    796.4009           1        671        3163

failures                    24834    .0688531           0          0           6
--------------------------------------------------------------------------------

. stsum

         failure _d:  failure50pc == 2
   analysis time _t:  (Delinq50pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

         |              incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+-------------------------------------------------------------------
   total |    287246677   .0000865        360681          .         .         .

. *-------------------------------------------------------------------------------
```

Resulting file; variable descriptions, incl. of calculated ones

```
Contains                                  data                                  from
C:\...\Analyses\Analyses110524BasicLoanFile\110524_2037AB_IndivLoans_work02.dta
  obs:       949,349
 vars:            90                          24 May 2011 20:52
 size:   285,754,049 (31.9% of memory free)
```

```
------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------
recno           float  %9.0g                 Record identifier
include         byte   %8.0g                Included if MFOrganizerGid and GroupGId both
                                             present
IsIndividualL~n long   %12.0g
LastObservati~e long   %tdD_m_Y              Date of last observation
IndividualCus~s str7   %7s
LoanGId         long   %12.0g                Loan GId (unique to this dataset)
IndividualC~GId long   %12.0g
GroupGId        long   %12.0g                Group GId
MFOrganizerGid  long   %12.0g
BranchGId       long   %12.0g
DisbursedAmt    float  %9.0g
BranchName      str21  %21s                  Branch
BranchGISId     str3   %3s
MFOrganizerID   str6   %6s
GroupCatgory    str18  %18s                  Type of borrower group
GroupGender     str15  %15s                  Gender of group members
GroupStatus     str7   %7s                   Group status
GroupId         str6   %6s                   Group ID
IndividualC~rId long   %12.0g
LoanStatus      str7   %7s                   Loan status
LoanId          str7   %7s                   Loan ID (within branch)
OpenDate        long   %tdD_m_Y              Date loan disbursed
MaturityDate    long   %tdD_m_Y              Date loan matured
PriAmtPaidAtE~s float   %9.0g                    Principal repaid by end of observation
period
OduePriAmtAtE~s float  %9.0g                  Overdue on principal by end of observation
period
LoanRepaidDate  long   %tdD_m_Y              Date loan repaid
DiffLastTrans~e float  %9.0g                 Delay (in days) betw. last cash transaction
                                              and formal replayment
BalanceOnMatu~e float  %9.0g
LastCashTrans~e long   %tdD_m_Y              Date of last cash transaction
WasDelinq01pc   byte   %8.0g                 Loan was delinquent > 01pc of principal
failure01pc     byte   %26.0g      failure    Failure event (01pc delinquent)
Delinq01pcOrE~e float  %tdD_m_Y              End of observation period (01pc delinq
                                                   model)
stset01pcdel    byte   %8.0g                 Loan sample of the 01pc delinquency model
Overlaps~1pcDel byte   %8.0g                 Obs period overlaps with next record (01pc
                                                   delinq model)
Delinq01pcOrE~j float  %tdD_m_Y                End of observation period (01pc - adj for
overlap)
stset01pcdelAdj byte    %8.0g                    Loan sample of the 01pc model (adj. for
overlap)
WasDelinq05pc   byte   %8.0g                 Loan was delinquent > 05pc of principal
failure05pc     byte   %26.0g      failure    Failure event (05pc delinquent)
Delinq05pcOrE~e float   %tdD_m_Y                  End of observation period (05pc delinq
model)
stset05pcdel    byte   %8.0g                 Loan sample of the 05pc delinquency model
Overlaps~5pcDel byte   %8.0g                  Obs period overlaps with next record (05pc
delinq model)
Delinq05pcOrE~j float  %tdD_m_Y                End of observation period (05pc - adj for
overlap)
stset05pcdelAdj byte    %8.0g                    Loan sample of the 05pc model (adj. for
overlap)
WasDelinq10pc   byte   %8.0g                 Loan was delinquent > 10pc of principal
failure10pc     byte   %26.0g      failure    Failure event (10pc delinquent)
Delinq10pcOrE~e float   %tdD_m_Y                  End of observation period (10pc delinq
model)
stset10pcdel    byte   %8.0g                 Loan sample of the 10pc delinquency model
Overlap~10pcDel byte   %8.0g                  Obs period overlaps with next record (10pc
delinq model)
Delinq10pcOrE~j float  %tdD_m_Y                End of observation period (10pc - adj for
overlap)
stset10pcdelAdj byte    %8.0g                    Loan sample of the 10pc model (adj. for
overlap)
WasDelinq20pc   byte   %8.0g                 Loan was delinquent > 20pc of principal
failure20pc     byte   %26.0g      failure    Failure event (20pc delinquent)
Delinq20pcOrE~e float   %tdD_m_Y                  End of observation period (20pc delinq
model)
stset20pcdel    byte   %8.0g                 Loan sample of the 20pc delinquency model
Overlap~20pcDel byte   %8.0g                  Obs period overlaps with next record (20pc
delinq model)
Delinq20pcOrE~j float  %tdD_m_Y                End of observation period (20pc - adj for
overlap)
stset20pcdelAdj byte    %8.0g                    Loan sample of the 20pc model (adj. for
overlap)
```

```
WasDelinq30pc   byte   %8.0g                 Loan was delinquent > 30pc of principal
failure30pc     byte   %26.0g      failure    Failure event (30pc delinquent)
Delinq30pcOrE~e float   %tdD_m_Y                 End of observation period (30pc delinq
model)
stset30pcdel    byte   %8.0g                 Loan sample of the 30pc delinquency model
Overlap~30pcDel byte   %8.0g                  Obs period overlaps with next record (30pc
delinq model)
Delinq30pcOrE~j float  %tdD_m_Y                End of observation period (30pc - adj for
overlap)
stset30pcdelAdj byte    %8.0g                    Loan sample of the 30pc model (adj. for
overlap)
WasDelinq40pc   byte   %8.0g                 Loan was delinquent > 40pc of principal
failure40pc     byte   %26.0g      failure    Failure event (40pc delinquent)
Delinq40pcOrE~e float   %tdD_m_Y                 End of observation period (40pc delinq
model)
stset40pcdel    byte   %8.0g                 Loan sample of the 40pc delinquency model
Overlap~40pcDel byte   %8.0g                  Obs period overlaps with next record (40pc
delinq model)
Delinq40pcOrE~j float  %tdD_m_Y                End of observation period (40pc - adj for
overlap)
stset40pcdelAdj byte    %8.0g                    Loan sample of the 40pc model (adj. for
overlap)
WasDelinq50pc   byte   %8.0g                 Loan was delinquent > 50pc of principal
failure50pc     byte   %26.0g      failure    Failure event (50pc delinquent)
Delinq50pcOrE~e float   %tdD_m_Y                 End of observation period (50pc delinq
model)
stset50pcdel    byte   %8.0g                 Loan sample of the 50pc delinquency model
Overlap~50pcDel byte   %8.0g                  Obs period overlaps with next record (50pc
delinq model)
Delinq50pcOrE~j float  %tdD_m_Y                End of observation period (50pc - adj for
overlap)
stset50pcdelAdj byte    %8.0g                    Loan sample of the 50pc model (adj. for
overlap)
FirstDateO~50Pc  long    %tdD_m_Y                   First date loan overdue over 50 pc
principal
FirstDateO~40Pc  long    %tdD_m_Y                   First date loan overdue over 40 pc
principal
FirstDateO~30Pc  long    %tdD_m_Y                   First date loan overdue over 30 pc
principal
FirstDateO~20Pc  long    %tdD_m_Y                   First date loan overdue over 20 pc
principal
FirstDateO~10Pc  long    %tdD_m_Y                   First date loan overdue over 10 pc
principal
FirstDateOD~5Pc long   %tdD_m_Y              First date loan overdue over 5 pc principal
FirstDateOD~1Pc long   %tdD_m_Y              First date loan overdue over 1 pc principal
_st             byte   %8.0g
_d              byte   %8.0g
_origin         int    %10.0g
_t              int    %10.0g
_t0             int    %10.0g
------------------------------------------------------------------------------------------
--------------------
Sorted by:  recno


. * Reminders:

. count
949349

. count if include
884126

. *       log:
C:\...\Analyses\Analyses110524BasicLoanFile\110524_2052AB_IndividLoanTable_to_stset.log
```

Samples sizes, dependent on delinquency model

See log file:
`C:\...\Analyses\Analyses110525LoanFile\110525_1118AB_IndivLoans_BasicStatsAndTags_CORR.log`

Example of sample flag labels:

```
              storage  display     value
variable name   type   format      label      variable label
```

```
------------------------------------------------------------------------------------------
-------------------
stset01pcdel    byte   %8.0g                 Loan sample of the 01pc delinquency model
stset01pcdelAdj byte   %8.0g                    Loan sample of the 01pc model (adj. for
overlap)
```

and accordingly for other models.

```
    variable |       sum
-------------+----------
stset01pcdel |    827058
stset01pcd~j |    881476
stset05pcdel |    795891
stset05pcd~j |    881430
stset10pcdel |    767483
stset10pcd~j |    881457
stset20pcdel |    756303
stset20pcd~j |    881433
stset30pcdel |    752636
stset30pcd~j |    881426
stset40pcdel |    750763
stset40pcd~j |    881419
stset50pcdel |    749360
stset50pcd~j |    881418
------------------------
```

Tagging loan records for borrowers, groups, organizers, branches

Tags

```
. * Tagging records for various entity levels:
. * As much as possible, loans tagged should be in the adjusted-for-overlap samples.
. * Since the 10pc delinquency level model is the leading one, we use this sample.
. * Also tags should be on first loans within a borrower.
.  gsort    BranchGId    MFOrganizerGid  GroupGId  IndividualCustomerGId    -stset10pcdelAdj
LoanGId

.
. * Branch
. egen tag branchtag = tag( BranchGISId)

. * Organizer
. egen tag MForgtag = tag( MFOrganizerGid)

.
. * Group
. egen grouptag = tag( GroupGId)

.
. * Individual customer:
. egen indivtag = tag( IndividualCustomerGId)


. tabstat *tag if  IsIndividualLoan, stat(sum) c(s)

    variable |       sum
-------------+----------
    indivtag |    405404
   branchtag |       150
    MForgtag |       912
    grouptag |     16027
------------------------

. tabstat *tag if  IsIndividualLoan &  stset10pcdelAdj, stat(sum) c(s)

    variable |       sum
-------------+----------
    indivtag |    360687
   branchtag |       150
    MForgtag |       912
    grouptag |     15874
------------------------

Reminder: File at this point is still:
C:\...\Analyses\Analyses110524BasicLoanFile\110524_2037AB_IndivLoans_work02.dta
```

Loan sequence within individual customers

```
.
. * Create loan sequence within each individual customer
. gsort  IndividualCustomerGId OpenDate

. by  IndividualCustomerGId: gen LoanSeq = _n

. label var LoanSeq "Sequence of loan for individual customer"

. gsort  IndividualCustomerGId  LoanSeq

. by  IndividualCustomerGId: egen NumberLoansByCustomer = max( LoanSeq)

. label var  NumberLoansByCustomer "Number of loans taken by customer"

. summ  NumberLoansByCustomer if  indivtag, detail


                    Number of loans taken by customer
-------------------------------------------------------------
      Percentiles      Smallest
 1%             1              1
 5%             1              1
10%             1              1       Obs              405404
25%             1              1       Sum of Wgt.      405404

50%             2                      Mean           2.341736
                        Largest       Std. Dev.       1.56725
75%             3             25
90%             5             25       Variance       2.456273
95%             5            132       Skewness       4.650647
99%             7            149       Kurtosis       309.1348
```

which later led to manual corrections of outliers.

Adjustments to consumer price index (CPI)

Monthly intra- and extrapolation

CPI data from Bangladesh Bank (2010).

1996 - 2006: only yearly values. Monthly values July 2006 - August 2010
1995-96 = 100 [base]

```
use C:\..\Analyses110102Thakur\monthlycpi.dta

-------------------------------------------------------------------------------------------

DateLastMonth                             Month and year (date value = last day of month)

                range:  [13179,18566]                units:  1
       or equivalently:  [31jan1996,31oct2010]       units:  days
         unique values:  178                      missing .:  0/178
-------------------------------------------------------------------------------------------

. ipolate  CPI DateLastMonth, gen(CPIintpol) e

. lowess  CPIintpol DateLastMonth, gen(CPIlowess) bwidth(0.1)
```

We work with this smoothed CPI variable. See graph in main report.

```
* 233.539 is the value for end of August 2010, the latest Bangladesh Bank value in
November 2010, when these smooths were computed.
```

Import into the loan table

Using Blasnik's *nearmrg* routine (Blasnik and Smith Undated)

```
use "C:\...\Analyses\Analyses110524BasicLoanFile\110524_2037AB_IndivLoans_work02.dta


. * Temporarily create auxiliary to match using file var DateLastMonth:
. gen DateLastMonth =  OpenDate

. sort  DateLastMonth

.    nearmrg    using    "C:\..\monthlycpi.dta",    nearvar(    DateLastMonth)    upper
genmatch(dateCPIused)

. tab  _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          3 |    949,349      100.00      100.00
------------+-----------------------------------
      Total |    949,349      100.00

. summ  CPIlowess

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
   CPIlowess |    949349    197.6412    20.87287   132.1868   236.5961
```

Calculation of adjusted loan principals

```
. gen  DisbursedAmtCPIadj =  DisbursedAmt * (233.539 /  CPIlowess)

. * re the value: 233.539, see above.

. gen logPrincipCPI = log10(DisbursedAmtCPIadj)
```

Comparison raw and CPI-adjusted loan principals

```
.
. summ  DisbursedAmt DisbursedAmtCPIadj logPrincipCPI, detail

                    Loan amount disbursed
-------------------------------------------------------------
      Percentiles      Smallest
 1%         2000             50
 5%         3000             50
10%         4000             50       Obs                 949349
25%         5000             50       Sum of Wgt.         949349

50%         6000                      Mean           7924.061
                        Largest       Std. Dev.      6465.692
75%        10000         300000
90%        13000         300000       Variance       4.18e+07
95%        16000         300000       Skewness       7.513738
99%        30000         400000       Kurtosis       137.5801

            Loan principal (adj. to CPI Sep 2010)
-------------------------------------------------------------
      Percentiles      Smallest
 1%     2373.626       51.44724
 5%     3560.439       52.07607
10%     4341.135       52.31993       Obs                 949349
25%     5636.343        53.0595       Sum of Wgt.         949349

50%     7665.347                      Mean           9263.284
                        Largest       Std. Dev.      7055.121
75%     10852.84       300000.6
90%     15171.34       322905.9       Variance       4.98e+07
95%     19374.35       328514.8       Skewness       6.928266
99%      34800.6       400000.8       Kurtosis       115.1421

        Loan principal (log10) - adj to CPI Sep 2010
-------------------------------------------------------------
      Percentiles      Smallest
 1%     3.375412       1.711362
 5%     3.551503       1.716638
10%     3.637603       1.718667       Obs                 949349
25%     3.750998       1.724763       Sum of Wgt.         949349
```

```
50%     3.884532                    Mean           3.899374
                      Largest       Std. Dev.      .2278381
75%     4.035543      5.477122
90%     4.181024      5.509076      Variance       .0519102
95%     4.287227      5.516555      Skewness       .4554251
99%     4.541587      5.602061      Kurtosis       4.812093
```

Detection of irregularities and action taken

Loans before global observation period

Marked but excluded later (see below).

```
log:
C:\...\Analyses\Analyses110526_1_Exclusions\110526_0925AB_MarkFirstLoansBefore0406113.log
```

Customers whose first loans were opened before global observation period, i.e., before 13 June 2004

```
. by IndividualCustomerGId: egen DateFirstLoanOpened = min(OpenDate)

. gen byte CustomStartBefore040613 = ( DateFirstLoanOpened < mdy(6,13,2004))

. tab  CustomStartBefore040613

CustomStart |
Before04061 |
          3 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    948,496       99.91       99.91
          1 |        853        0.09      100.00
------------+-----------------------------------
      Total |    949,349      100.00

. count if  CustomStartBefore040613 & indivtag
  228

. count if  CustomStartBefore040613 &  MForgtag
    0

. count if  CustomStartBefore040613 &   grouptag
   13

. * i.e., subsequently had to recalculate group tags.
```

Customers with first and second loans opened on the same date

```
       log:
C:\...\Analyses\Analyses110526_1_Exclusions\110526_0946AB_TestForSecondLoanSameDateAsFirs
t.log


. gsort  IndividualCustomerGId  LoanSeq

. gen byte FirstSecLoansSameOpenDate = 0

. replace  FirstSecLoansSameOpenDate = 1 if LoanSeq == 1 & OpenDate == OpenDate[_n+1] &
IndividualCustomerGId == IndividualCustomerGId[_n+1]

. tab indivtag if  FirstSecLoansSameOpenDate

tag(Individ |
ualCustomer |
       GId) |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        622       70.36       70.36
          1 |        262       29.64      100.00
------------+-----------------------------------
      Total |        884      100.00

. * Probably ok; tags may have gone to second loans in this sort order because the sort
order during tagging
> had been different among loans of same opening date.
```

```
. * Check now for number loans for such customers:

. tab  NumberLoansByCustomer if  FirstSecLoansSameOpenDate, missing

  Number of |
loans taken |
by customer |      Freq.     Percent        Cum.
------------+-----------------------------------
          2 |        295       33.37       33.37
          3 |        217       24.55       57.92
          4 |        128       14.48       72.40
          5 |         95       10.75       83.14
          6 |         64        7.24       90.38
          7 |         47        5.32       95.70
          8 |         17        1.92       97.62
          9 |          4        0.45       98.08
         10 |          3        0.34       98.42
         11 |          2        0.23       98.64
         12 |          1        0.11       98.76
         13 |          1        0.11       98.87
         14 |          2        0.23       99.10
         15 |          2        0.23       99.32
         17 |          1        0.11       99.43
         18 |          1        0.11       99.55
         25 |          2        0.23       99.77
        132 |          1        0.11       99.89
        149 |          1        0.11      100.00
------------+-----------------------------------
      Total |        884      100.00

. * Inspect some, e.g. those who took 25 loans:

.  list    IndividualCustomerGId  OpenDate  DisbursedAmt  if    CustFirstSecSameDate  &
NumberLoansByCustomer == 25

        +---------------------------------+
        | Indi~GId    OpenDate   Disbur~t |
        |---------------------------------|
343231. |   147150   20 Dec 05       2000 |
343232. |   147150   20 Dec 05       2000 |
343233. |   147150   20 Dec 05       2000 |
343234. |   147150   20 Dec 05       2000 |
343235. |   147150   20 Dec 05       2000 |
        |---------------------------------|
343236. |   147150   20 Dec 05       2000 |
343237. |   147150   20 Dec 05       2000 |
343238. |   147150   20 Dec 05       2000 |
343239. |   147150   20 Dec 05       2000 |
343240. |   147150   20 Dec 05       2000 |
        |---------------------------------|
343241. |   147150   20 Dec 05       2000 |
343242. |   147150   20 Dec 05       2000 |
343243. |   147150   20 Dec 05       2000 |
343244. |   147150   20 Dec 05       2000 |
343245. |   147150   20 Dec 05       2000 |
        |---------------------------------|
343246. |   147150   20 Dec 05       2000 |
343247. |   147150   20 Dec 05       2000 |
343248. |   147150   20 Dec 05       2000 |
343249. |   147150   20 Dec 05       2000 |
343250. |   147150   20 Dec 05       2000 |
        |---------------------------------|
343251. |   147150   20 Dec 05       2000 |
343252. |   147150   20 Dec 05       2000 |
343253. |   147150   20 Dec 05       2000 |
343254. |   147150   20 Dec 05       2000 |
343255. |   147150   20 Dec 05       2000 |
--Break--

. * These obviously are group loans.
```

Exclusions

New working file:

```
.                                                                                save
"C:\...\Analyses\Analyses110526_1_Exclusions\110526_1111AB_IndivLoans_Jun2004_Sep2010_wor
k03.dta"
```

Borrowers who took out their first loan before global observation period

```
. * Exclude loans of borrower who took out first loan before 13 June 2004 (= first loan
opened in Thakurgaon):

. tab  CustomStartBefore040613

CustomStart |
Before04061 |
          3 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    948,496       99.91       99.91
          1 |        853        0.09      100.00
------------+-----------------------------------
      Total |    949,349      100.00

. count if indivtag &  CustomStartBefore040613
  228

. count if grouptag &  CustomStartBefore040613
   13

. count if MForgtag &  CustomStartBefore040613
    0

. count if branchtag &  CustomStartBefore040613
    0

. * i.e., will need to recalculate group tags

. drop if  CustomStartBefore040613
(853 observations deleted)
```

Some of the cases with first and second loan issued on the same date

```
. * Drop loans and customers if first and second loans issued same date and some
additional criteria (e.g. from inspection):

[after manual inspections:]
. drop if  CustFirstSecSameDate &  NumberLoansByCustomer > 14
(396 observations deleted)
as well as some individual customers with 14 or fewer loans
[for details see log file

resulting in
. count
948032

. * Distribution of borrowers by number of loans:

. tab  NumberLoansByCustomer if indivtag

  Number of |
loans taken |
by customer |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |    160,014       39.49       39.49
          2 |    100,975       24.92       64.42
          3 |     64,114       15.82       80.24
          4 |     36,222        8.94       89.18
          5 |     23,931        5.91       95.09
          6 |     13,076        3.23       98.31
          7 |      4,954        1.22       99.54
          8 |      1,276        0.31       99.85
          9 |        370        0.09       99.94
         10 |        110        0.03       99.97
         11 |         62        0.02       99.99
         12 |         32        0.01       99.99
         13 |         22        0.01      100.00
         14 |          3        0.00      100.00
         17 |          1        0.00      100.00
------------+-----------------------------------
      Total |    405,162      100.00

Case with 17 loans: manually inspected, was genuine.
```

Recalculating group tags

```
. * Sort order earlier used for tagging was:
```

```
. gsort   BranchGId   MFOrganizerGid  GroupGId  IndividualCustomerGId   -stset10pcdelAdj
LoanGId

. ren grouptag grouptagold

. help egen

. egen grouptag = tag(GroupGId)

. tab grouptagold grouptag

tag(GroupG |     tag(GroupGId)
       Id) |         0          1 |     Total
-----------+----------------------+----------
         0 |   932,006         12 |   932,018
         1 |         0     16,014 |    16,014
-----------+----------------------+----------
     Total |   932,006     16,026 |   948,032


. tab grouptag

tag(GroupGI |
         d) |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    932,006       98.31       98.31
          1 |     16,026        1.69      100.00
------------+-----------------------------------
      Total |    948,032      100.00
```

Tags were again corrected later, to ensure that the customer tag would uniformly be in her first-loan record[1].

Import of administrative unit names

```
. sort BranchName

. merge  BranchName using
"C:\...\Data\MF\110526_BranchCodes\110526_1430AB_BranchAndAdminNamesUpdated.dta"
variable BranchName does not uniquely identify observations in the master data

.
. tab _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          2 |          4        0.00        0.00
          3 |    948,032      100.00      100.00
------------+-----------------------------------
      Total |    948,036      100.00


. drop if _merge == 2

Examples:

. gsort Unitname AreaOffice BranchName

. list  BranchName  AreaOffice Upazila District Unitname YearFounded if branchtag

      +-------------------------------------------------------------------------------------------+
      |            BranchName     AreaOffice           Upazila      District      Unitname   YearFo~d |
      |-------------------------------------------------------------------------------------------|
 1310. |      CDP_01_Begumganj    CDPKurigram            Ulipur      Kurigram   CDPKurigram       1991 |
 7876. |     CDP_02_Narayanpur    CDPKurigram        Nageshwari      Kurigram   CDPKurigram       1991 |
11687. |   CDP_03_Shaheberalga    CDPKurigram            Ulipur      Kurigram   CDPKurigram       1991 |
21231. |       Chi_01_Chilmari       Chilmari          Chilmari      Kurigram   CDPKurigram       1991 |
26532. |       Chi_02_Raniganj       Chilmari          Chilmari      Kurigram   CDPKurigram       2005 |
      |-------------------------------------------------------------------------------------------|
31193. |    Chi_03_Bohorarvita       Chilmari          Chilmari      Kurigram   CDPKurigram       2007 |
36548. |     Chi_04_Manushmara       Chilmari          Chilmari      Kurigram   CDPKurigram       1991 |
48294. |       Rji_01_Rajibpur       Rajibpur     Char Rajibpur      Kurigram   CDPKurigram       1991 |
```

[1] The details, recorded in C:\...\Analyses110614_2_RecalculateTags\110614_1252AB_RecalculateTags.log, are not of immediate concern to users of the final master tables. The resulting file initially was 110614_1302AB_IndivLoans_Jun2004_Sep2010_tagsCORR_work07.dta.

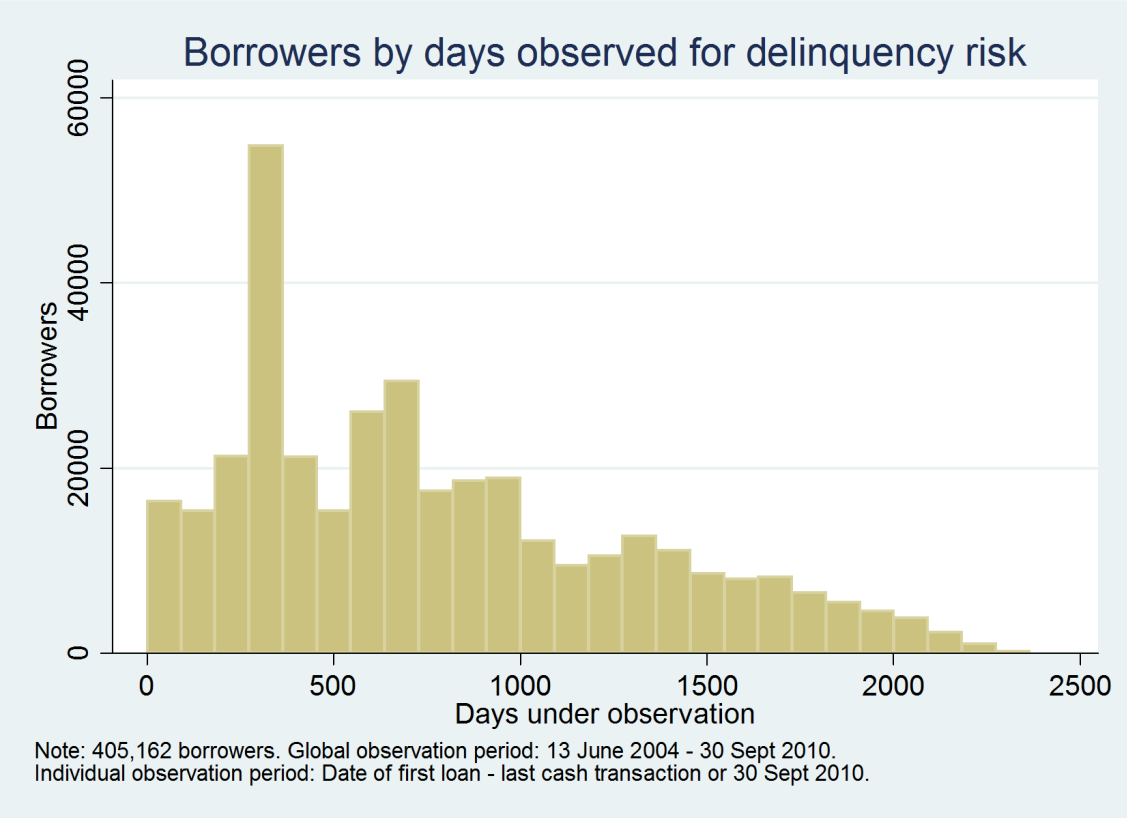
```
53864. |      Rji_02_Kodalkati       Rajibpur     Char Rajibpur      Kurigram   CDPKurigram       1991 |
58575. |    Rji_03_Kauniarchar       Rajibpur          Dewanganj     Jamalpur   CDPKurigram       2004 |
       |-------------------------------------------------------------------------------------------|
```

Towards survival analysis: Days under observation

```
      log:
C:\...\Analyses\Analyses110526_1_Exclusions\110526_1223AB_TimeUnderObservation.log

use
"C:\...\Analyses\Analyses110526_1_Exclusions\110526_1111AB_IndivLoans_Jun2004_Sep2010_wor
k03.dta"

. * Time form open date first loan to latest cash transaction date across all loans in
given borrower:

. * [Earlier had calculated:

.     *     bysort     IndividualCustomerGId:     egen     LastCashTrDateByBorrower     =
max(LastCashTransactionDate)]

. gen DaysFrFirstLoanToLastTrans =  LastCashTrDateByBorrower - _origin

 [_origin is a result of stset and is defined for the st sample]
. summ  DaysFrFirstLoanToLastTrans if indivtag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
DaysFrFirs~s |    359151    779.2536    527.7897          1       2300

.
. count if  DaysFrFirstLoanToLastTrans ==. &  indivtag
46011

.
. * Define end of observation as the smaller of LastCashTrDateByBorrower or 30 Sep 2010
[the latter kicks in only if the former missing]

. gen temp = mdy(9,30,2010)

. egen EndObsPeriod = rowmin( LastCashTrDateByBorrower temp)

. drop temp

. gen DaysUnderObs =  EndObsPeriod - _origin

. label var  DaysUnderObs "Days under observation"

. label var  EndObsPeriod "Date last cash transact by borrower. If missing, 30 Sep 2010"

. format  %tdD_m_Y  EndObsPeriod

. codebook  EndObsPeriod DaysUnderObs  if indivtag

-------------------------------------------------------------------------------------------
EndObsPeriod                Date last cash transact by borrower. If missing, 30 Sep 2010
-------------------------------------------------------------------------------------------

                  type:  numeric daily date (float)

                 range:  [16279,18535]                units:  1
       or equivalently:  [27jul2004,30sep2010]        units:  days
         unique values:  1681                     missing .:  0/405162

                  mean:   18143.5 = 03sep2009 (+ 12 hours)
             std. dev:   506.518

          percentiles:        10%       25%       50%       75%       90%
                           17286     17865     18413     18532     18534
                         30apr2007 29nov2008 31may2010 27sep2010 29sep2010

-------------------------------------------------------------------------------------------
DaysUnderObs
Days under observation
-------------------------------------------------------------------------------------------

                  type:  numeric (float)
```

```
        range:  [0,2300]                    units:  1
unique values:  2301                    missing .:  44540/405162

         mean:   776.362
      std. dev:   529.027

  percentiles:        10%       25%       50%       75%       90%
                      203       333       662      1118      1595
```

Figure 2: Borrowers by days of risk of delinquency - histogram



Recoding of borrower group categories and gender

```
log:
C:\...\Analyses\Analyses110526_1_Exclusions\110526_1545AB_RecodeGroupCategoriesAndGender.
log
```

Group categories

Recodings

```
. tab  GroupCatgory if grouptag, missing

  Type of borrower |
             group |      Freq.     Percent        Cum.
-------------------+-----------------------------------
                   |        159        0.99        0.99
            Active |         71        0.44        1.44
               DSF |         45        0.28        1.72
          Existing |        873        5.45        7.16
        Graduated  |      1,871       11.67       18.84
               HCP |      1,048        6.54       25.38
                LM |      6,146       38.35       63.73
              PLDP |          2        0.01       63.74
           Primary |        982        6.13       69.87
```

```
            Prime |      2,289       14.28       84.15
             RDRS |         87        0.54       84.69
              RMC |         66        0.41       85.11
              SFG |      1,263        7.88       92.99
        Secondary |        377        2.35       95.34
           Tribal |        212        1.32       96.66
       Ultra-poor |        529        3.30       99.96
              VGD |          6        0.04      100.00
------------------+-----------------------------------
            Total |     16,026      100.00
```

Subsequently, these group categories were further reduced. For each of the reduced types, the typical poverty level of borrowers at entry was determined by a group of Microfinance Program senior managers. The poverty at entry level through the program that recruited the borrower is the closest thing to a household at baseline poverty measurement available to this study.

The reduction and recoding followed this table:

GroupCategoryEnc	GroupCategoryFirstRecode	GroupCatReduced	PovertyAtEntry
1	Active	Landless and marginal	2-Poor
2	DSF	Small farmer group	1-Middle class
3	Existing	Landless and marginal	2-Poor
4	Graduated	Landless and marginal	2-Poor
5	HCP	Ultra-poor	3-Ultra-poor
6	LM	Landless and marginal	2-Poor
7	PLDP	Ultra-poor	3-Ultra-poor
8	Primary	Landless and marginal	2-Poor
9	Prime	Ultra-poor	3-Ultra-poor
10	RDRS	Landless and marginal	2-Poor
11	RMC	Landless and marginal	2-Poor
12	SFG	Small farmer group	1-Middle class
13	Secondary	Landless and marginal	2-Poor
14	Tribal	Tribal	3-Ultra-poor
15	Ultra-poor	Ultra-poor	3-Ultra-poor
16	VGD	Ultra-poor	3-Ultra-poor

The look-up table in STATA holds these encoded variables:

```
-------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------
GroupCategory~c byte   %8.0g                  GroupCategoryEnc
GroupCatReduc~c long   %21.0g      GroupCatReducEnc
                                              Group type (reduced)
PovertyAtEntr~c long   %14.0g      PovertyAtEntryEnc
                                        Implied household poverty level at program entry
-------------------------------------------------------------------------------------------

. label list
PovertyAtEntryEnc:
           1 Middle class
           2 Poor
           3 Ultra-poor

GroupCatReducEnc:
           1 Landless and marginal
```

```
           2 Small farmer group
           3 Tribal
           4 Ultra-poor
```

Merging into loan table

```
log:
C:\...\Analyses\Analyses110615_1_GroupsRecoded\110615_1255AB_GroupsRecoded2MasterTables.l
og

. * 1. Recode in unsplit loan file table:

use"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep
2010_tagsCORR_work07.dta

. sort  GroupCategoryEnc

.               merge                                GroupCategoryEnc               using
C:\...\Analyses\Analyses110615_1_GroupsRecoded\110615_1246AB_GroupsRecodedLookupTable.dta

. tab _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |     71,649        7.56        7.56
          3 |    876,383       92.44      100.00
------------+-----------------------------------
      Total |    948,032      100.00
```

and similarly:

```
. * 2. Recode in episode-split loan file table;

.
use
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1549AB_IndivLoans_10pcDeli
nq_EpisodeSplit_work06.dta"

. sort  GroupCategoryEnc

. merge  GroupCategoryEnc using C:\...\Analyses\Analyses110615_1_GroupsRecoded
> \110615_1246AB_GroupsRecodedLookupTable.dta
variable GroupCategoryEnc does not uniquely identify observations in the master data

. tab _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |     32,681        0.95        0.95
          3 |  3,391,099       99.05      100.00
------------+-----------------------------------
      Total |  3,423,780      100.00

In both tables, dummies were generated for each poverty at entry level:

. gen byte PovEntryMiddle = ( PovertyAtEntryEnc==1)

. gen byte PovEntryPoor = ( PovertyAtEntryEnc==2)

. gen byte PovEntryUltra = ( PovertyAtEntryEnc==3)

. gen byte PovEntryMissing = ( PovertyAtEntryEnc==.)

. label var  PovEntryMiddle  "Implied poverty level at entry: middle class"

. label var   PovEntryPoor"Implied poverty level at entry: poor"

. label var    PovEntryUltra "Implied poverty level at entry: ultra-poor"

. label var     PovEntryMissing "Implied poverty level at entry: undefined"
```

Gender

```
. gen byte GroupIsFemale = 1

. replace  GroupIsFemale = 0 if GroupGender=="Male"
```

```
(110225 real changes made)

. replace  GroupIsFemale = .a if GroupGender=="Unknown"
(66552 real changes made, 66552 to missing)

. replace  GroupIsFemale = .b if GroupGender=="FederationMixed"
(572 real changes made, 572 to missing)

.
. move  GroupIsFemale GroupGender

. tab GroupIsFemale, missing

GroupIsFema |
         le |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    110,225       11.63       11.63
          1 |    770,683       81.29       92.92
         .a |     66,552        7.02       99.94
         .b |        572        0.06      100.00
------------+-----------------------------------
      Total |    948,032      100.00

. tab GroupIsFemale if grouptag, missing

GroupIsFema |
         le |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      2,097       13.08       13.08
          1 |     13,721       85.62       98.70
         .a |        193        1.20       99.91
         .b |         15        0.09      100.00
------------+-----------------------------------
      Total |     16,026      100.00

. drop  GroupGender
```

Episode splitting to accommodate time-varying covariates

Using
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1521AB_IndivLoans_Jun2004_Sep2010_work04.dta"

Exclusions

Drop records without group or organizer ID

```
. tab include

Included if |
MFOrganizer |
    Gid and |
   GroupGId |
       both |
    present |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     65,110        6.87        6.87
          1 |    882,922       93.13      100.00
------------+-----------------------------------
      Total |    948,032      100.00

. drop if include == 0
```

Work with 10 percent delinquency model

```
. * reset for the 10pc delinquency level model:

.   stset   Delinq10pcOrEnddateAdj   ,   id(IndividualCustomerGId)   failure(failure10pc==2)
time0(OpenDate) exit(time .) or
> igin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure10pc == 2
obs. time interval:  (OpenDate, Delinq10pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
```

```
            origin:  time OpenDate

--------------------------------------------------------------------------------
   882922  total obs.
     2261  entry on or after exit (OpenDate>Delinq10pcOrEnddateAdj)PROBABLE ERROR
--------------------------------------------------------------------------------
   880661  obs. remaining, representing
   360492  subjects
   170469  failures in multiple failure-per-subject data
 2.41e+08  total analysis time at risk, at risk from t =         0
                              earliest observed entry t =         0
                                   last observed exit t =      2300

.                                                                                save
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1522AB_IndivLoans_Jun2004_
Sep2010_small_work05.dta", replace
file

. stdes

         failure _d:  failure10pc == 2
   analysis time _t:  (Delinq10pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

                                  |-------------- per subject --------------|
Category                    total        mean         min     median        max
--------------------------------------------------------------------------------
no. of subjects            360492
no. of records             880661    2.442942           1          2         14

(first) entry time                    .0570748           0          0        372
(final) exit time                     731.3168           1        608       2300

subjects with gap          206403
time on gap if gap       22670575    56.30203           1         11       2006
time at risk            2.409e+08    668.3719           1        573       2298

failures                   170469    .4728787           0          0          9
--------------------------------------------------------------------------------

. stsum

         failure _d:  failure10pc == 2
   analysis time _t:  (Delinq10pcOrEnddateAdj-origin)
             origin:  time OpenDate
  exit on or before:  time .
                 id:  IndividualCustomerGId

         |               incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+----------------------------------------------------------------------
   total |    240942720   .0007075        360492        472       928      1735
```

Reduce to minimum set of variables

for file size considerations

```
. keep  recno U_id D_ID BranchGId MFOrganizerGid GroupCategoryEnc GroupIsFemale GroupGId
IndividualCustomerGId     LoanGId          OpenDate     DisbursedAmtCPIadj     failure10pc
Delinq10pcOrEnddateAdj _st _d _origin _t _t0

.                                                                                save
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1522AB_IndivLoans_Jun2004_
Sep2010_small_work05.dta", replace

--------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------
recno           float  %9.0g                 Record identifier
U_id            byte   %8.0g                 RDRS Unit identifier
D_ID            byte   %8.0g                 District Unit identifier
BranchGId       int    %12.0g                Branch ID
MFOrganizerGid  int    %12.0g                Organizer ID
GroupCategory~c byte   %10.0g      GroupCategoryEnc
                                             Type of borrower group
GroupIsFemale   byte   %8.0g                 Borrower group is female
GroupGId        int    %12.0g                Group GId
```

```
IndividualCus~d long   %12.0g                 Customer ID
LoanGId         long   %12.0g                 Loan GId (unique to this dataset)
OpenDate        int    %tdD_m_Y               Date loan disbursed
DisbursedAmtC~j float  %9.0g                  Loan principal (adj. to CPI Sep 2010)
failure10pc     byte   %26.0g      failure    Failure event (10pc delinquent)
Delinq10pcOrE~j int    %tdD_m_Y               End of observation period (10pc - adj for
                                                    overlap)

_st             byte   %8.0g
_d              byte   %8.0g
_origin         int    %10.0g
_t              int    %10.0g
_t0             int    %10.0g
------------------------------------------------------------------------------------------
-------------------------
Sorted by:  recno

. count
882922
```

Determine split categories

Quarter-yearly splits, with 13 June - 30 Sept 2004 combined into the first:

PeriodEndDate	STATAdaynumbering	DaysAfter12June2004	UseForSplit
30-Jun-04	16252	18	0
30-Sep-04	16344	110	111
31-Dec-04	16436	202	203
31-Mar-05	16526	292	293
30-Jun-05	16617	383	384
30-Sep-05	16709	475	476
31-Dec-05	16801	567	568
31-Mar-06	16891	657	658
30-Jun-06	16982	748	749
30-Sep-06	17074	840	841
31-Dec-06	17166	932	933
31-Mar-07	17256	1022	1023
30-Jun-07	17347	1113	1114
30-Sep-07	17439	1205	1206
31-Dec-07	17531	1297	1298
31-Mar-08	17622	1388	1389
30-Jun-08	17713	1479	1480
30-Sep-08	17805	1571	1572
31-Dec-08	17897	1663	1664
31-Mar-09	17987	1753	1754
30-Jun-09	18078	1844	1845
30-Sep-09	18170	1936	1937
31-Dec-09	18262	2028	2029
31-Mar-10	18352	2118	2119
30-Jun-10	18443	2209	2210
30-Sep-10	18535	2301	

Episode splitting

```
. stsplit splitcat, after(time=mdy(6,12,2004)) at(0 111 203 293 384 476 568 658 749 841
933 1023 1114 1206 1298 1389 1480 1572 1664 1754 1845 1937 2029 2119 2210)
(2540858 observations (episodes) created)

.                                                                                save
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1549AB_IndivLoans_10pcDeli
nq_EpisodeSplit_work06.dta", replace
```

resulting in a file of 144 MB. Episodes are distributed over the split categories as in

```
. count
3423780

. tab _st

        _st |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      2,261        0.07        0.07
          1 |  3,421,519       99.93      100.00
------------+-----------------------------------
      Total |  3,423,780      100.00

. tab splitcat

   splitcat |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      3,954        0.12        0.12
        111 |     10,503        0.31        0.42
        203 |     21,739        0.64        1.06
        293 |     33,981        0.99        2.05
        384 |     49,075        1.43        3.49
        476 |     66,152        1.93        5.42
        568 |     81,837        2.39        7.81
        658 |     95,046        2.78       10.59
        749 |    108,921        3.18       13.77
        841 |    123,851        3.62       17.39
        933 |    136,382        3.99       21.38
       1023 |    144,363        4.22       25.60
       1114 |    150,345        4.39       29.99
       1206 |    151,658        4.43       34.42
       1298 |    163,477        4.78       39.20
       1389 |    176,283        5.15       44.35
       1480 |    189,185        5.53       49.88
       1572 |    201,022        5.88       55.76
       1664 |    218,463        6.38       62.14
       1754 |    215,732        6.31       68.45
       1845 |    206,881        6.05       74.49
       1937 |    211,816        6.19       80.69
       2029 |    216,570        6.33       87.02
       2119 |    218,582        6.39       93.40
       2210 |    225,701        6.60      100.00
------------+-----------------------------------
      Total |  3,421,519      100.00

. egen splitseq = group( splitcat)

. summ  splitseq

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    splitseq |   3421519    16.71111    5.791937          1         25
```

Importing time-varying overdue rates

Preparation of overdue rate tables

Overdue-on-principal and on-time-recovery rates were calculated for each group, organizer, and branch, for each month in which there were transactions under the concerned entity. The month-year, overdue and OTR values are held in three separate tables, one for groups, one for organizers, one for branches.

These values were imported to the one-record-per loan as well as to episode-split loan tables. While the match on group, organizer, respectively on branch has to be exact, the match on the period end date is a nearest value match. The causal direction investigated is from higher-entity overdue rate to individual-borrower delinquency hazard. Therefore, we import the nearest preceding value *prior* to loan opening date (unsplit table), respectively to the splitcat date (episode-split).

The STATA user-written command *nearmrg* (Blasnik and Smith Undated: see above) facilitates this. However, the values of the nearest-match variable in the to-be-imported table (the "using" table in STATA terms) have to be unique. Since most dates recur between entities in the same table, the uniqueness has to be created artificially. STATA must store values to the required precision such that they remain unique. This requires storage in the double format.

We demonstrate this for the branch-level variables[2].

```
. use
"C:\...\Analyses\Analyses110607_1_NearmrgExperim\110607_1305AB_BranchOverdue_MatchDateAsD
ouble.dta"

  obs:        14,446
-------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------
BranchGId       long   %12.0g
OnTimeRecover~e float  %9.0g
OverduePercOf~l float  %9.0g
-------------------------------------------------------------------------------------------

. gen double nearmatchdate =  PeriodEndDate - 1/_n

. codebook  nearmatchdate
-------------------------------------------------------------------------------------------
                  type:  numeric (double)

                 range:  [13756,18717]               units:  1.000e-08
         unique values:  14446                   missing .:  0/14446
-------------------------------------------------------------------------------------------

. drop  PeriodEndDate

. gsort  BranchGId nearmatchdate
. save
```

Note: "- 1 / _n" worked here as a jittering term, but something like "- _n / (_N + 1)" might be safer in the general case of integer variables to be made temporarily unique.

Merging into loan tables

Again, demonstrating for the branch level values:

Unsplit loan table

```
use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"
```

[2] For the group and organizer-level variables we depended on an inefficient, though correct work-around via Excel. STATA's own devices are preferrable.

```
clonevar nearmatchdate = OpenDate

. gsort    BranchGId nearmatchdate

.
.           nearmrg                                                 BranchGId           using
C:\...\Data\MF\110607_Aldo_OverdLevels\110607_1305AB_BranchOverdue_MatchDateAsDouble.dta  ,
lower genmatch( nearmatchdateUsed) nearvar( nearmatchdate)

. tab  _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |      5,743        0.61        0.61
          3 |    942,289       99.39      100.00
------------+-----------------------------------
      Total |    948,032      100.00

. ren  OnTimeRecoveryRate BranchOnTimeRecoveryRate

. label var   BranchOnTimeRecoveryRate "Branch level on time recovery rate before loan
opened"

. ren  OverduePercOfPrincipal BranchOverduePercOfPrincipal

. label var   BranchOverduePercOfPrincipal "Branch level overdue pc of principal before
loan opened"

. ren  nearmatchdateUsed BranchPeriodEndDateUsed

. ren  _merge _mergeBranchData
```

Episode-split loan table

```
.                                                                                   use
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1549AB_IndivLoans_10pcDeli
nq_EpisodeSplit_work06.dta"

. * Create a date at the beginning of the quarter-year period to which the overdue values
can be imported with the procedure nearmrg.

. * nearmatchdate is 12 June 2004 + splitcat
gen nearmatchdate = mdy(6,12,2004) +  splitcat

. * Insert branch values:

. gsort  BranchGId nearmatchdate

.           nearmrg                                                 BranchGId           using
"C:\...\Data\MF\110607_Aldo_OverdLevels\110607_1305AB_BranchOverdue_MatchDateAsDouble.dta
", nearvar( nearmatchdate) genmatch(PeriodEndBranch) lower

. * Note: The option "lower" ensures that the value from the nearest previous period is
selected for import, compared to the "nearmatchdate" in the loan table.

. tab _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |      8,733        0.26        0.26
          3 |  3,415,047       99.74      100.00
------------+-----------------------------------
      Total |  3,423,780      100.00

. ren  _merge mergeBranchOverd /// * Of potential use for in-sample designation.

. * Testing how many of the imported values are older than previous month:

. gen double TimeDiff =  nearmatchdate -  PeriodEndBranch

. ren  TimeDiff TimeDiffNearmrgBranch

. label var   TimeDiffNearmrgBranch "Days between loan episode start and end previous
month Branch overd"

. count if  TimeDiff > 31
27236
```

```
. count if  TimeDiff . 31 & TimeDiff ~= . /// [excludes the non-matches]
18503

. ren  OnTimeRecoveryRate BranchOTRbeginquart

. ren  OverduePercOfPrincipal BranchOvdBeginquart

drop PeriodEndBranch
```

Creating binaries for levels of overdue

The resulting files carry the time-variant overdue and OTR rates for groups, organizers and branches. We have subsequently used only overdue rates in this study.

Due to the highly skewed distributions of the overdue levels, we created binaries in both tables as follows:

```
. gen byte OverdGr1pcOD = ( GroupOverduePercOfPrincipal > 0.01 )

. replace  OverdGr1pcOD = . if  GroupOverduePercOfPrincipal ==.

.
. gen byte OverdGr2pcOD = ( GroupOverduePercOfPrincipal > 0.02 &
GroupOverduePercOfPrincipal ~= .)

. replace  OverdGr2pcOD = . if  GroupOverduePercOfPrincipal ==.

.
. gen byte OverdGr4pcOD = ( GroupOverduePercOfPrincipal > 0.04 &
GroupOverduePercOfPrincipal ~= .)

. replace  OverdGr4pcOD = . if  GroupOverduePercOfPrincipal ==.

.
. gen byte OverdGr8pcOD = ( GroupOverduePercOfPrincipal > 0.08 &
GroupOverduePercOfPrincipal ~= .)

. replace  OverdGr8pcOD = . if  GroupOverduePercOfPrincipal ==.
.
.
. gen byte OverdMFO1pcOD = (  MFOrgOverduePercOfPrincipal> 0.01 &
MFOrgOverduePercOfPrincipal ~= .)

. replace   OverdMFO1pcOD= . if   MFOrgOverduePercOfPrincipal ==.

.
. gen byte OverdMFO2pcOD = (  MFOrgOverduePercOfPrincipal> 0.02 &
MFOrgOverduePercOfPrincipal ~= .)

. replace   OverdMFO2pcOD= . if   MFOrgOverduePercOfPrincipal ==.

.
. gen byte OverdMFO4pcOD = (  MFOrgOverduePercOfPrincipal> 0.04 &
MFOrgOverduePercOfPrincipal ~= .)

. replace   OverdMFO4pcOD= . if   MFOrgOverduePercOfPrincipal ==.

.
. gen byte OverdBr2pcOD = (   BranchOverduePercOfPrincipal> 0.025 &
BranchOverduePercOfPrincipal ~= .)

. replace    OverdBr2pcOD = . if    BranchOverduePercOfPrincipal ==.

.
. gen byte OverdBr5pcOD = (   BranchOverduePercOfPrincipal> 0.05 &
BranchOverduePercOfPrincipal ~= .)

. replace     OverdBr5pcOD = . if     BranchOverduePercOfPrincipal ==.
```

The effects therefore have to be interpreted cumulatively.

Delinquency at loan and borrower levels

```
      log:
C:\...\Analyses\Analyses110615_1_GroupsRecoded\110615_1851AB_BorrowerEverDelinquentAtGive
nLevel.log
.
use"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep
2010_tagsCORR_work07.dta"
```

Descriptive statistics: The delinquency pyramid

Loan level dummies for "ever delinquent"

```
. * [already existing]
. des WasDe*
-------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------
WasDelinq01pc   byte   %8.0g                  Loan was delinquent > 01pc of principal
WasDelinq05pc   byte   %8.0g                  Loan was delinquent > 05pc of principal
WasDelinq10pc   byte   %8.0g                  Loan was delinquent > 10pc of principal
WasDelinq20pc   byte   %8.0g                  Loan was delinquent > 20pc of principal
WasDelinq30pc   byte   %8.0g                  Loan was delinquent > 30pc of principal
WasDelinq40pc   byte   %8.0g                  Loan was delinquent > 40pc of principal
WasDelinq50pc   byte   %8.0g                  Loan was delinquent > 50pc of principal
-------------------------------------------------------------------------------------------
```

At borrower level

```
. sort  IndividualCustomerGId

. by IndividualCustomerGId: egen CustomerEverDelinq01pc = max( WasDelinq01pc)

. label var  CustomerEverDelinq01pc "Customer was at some point delinquent with 1 pc of
principal"

. by IndividualCustomerGId: egen CustomerEverDelinq05pc = max( WasDelinq05pc)

. label var  CustomerEverDelinq05pc "Customer was at some point delinquent with 5 pc of
principal"

. by IndividualCustomerGId: egen CustomerEverDelinq10pc = max( WasDelinq10pc)

. label var  CustomerEverDelinq10pc "Customer was at some point delinquent with 10 pc of
principal"

. by IndividualCustomerGId: egen CustomerEverDelinq20pc = max( WasDelinq20pc)

. label var  CustomerEverDelinq20pc "Customer was at some point delinquent with 20 pc of
principal"

. by IndividualCustomerGId: egen CustomerEverDelinq30pc = max( WasDelinq30pc)

. label var  CustomerEverDelinq30pc "Customer was at some point delinquent with 30 pc of
principal"

. by IndividualCustomerGId: egen CustomerEverDelinq40pc = max( WasDelinq40pc)

. label var  CustomerEverDelinq40pc "Customer was at some point delinquent with 40 pc of
principal"

. by IndividualCustomerGId: egen CustomerEverDelinq50pc = max( WasDelinq50pc)

. label var  CustomerEverDelinq50pc "Customer was at some point delinquent with 50 pc of
principal"
```

Comparison loan and borrower level incidence

Loans

```
. tabstat WasD*, s(mean) c(s)

    variable |      mean
-------------+----------
```

```
WasDelin~1pc |  .5604927
WasDelin~5pc |  .3513953
WasDelinq1~c |  .1987391
WasDelinq2~c |  .1013489
WasDelinq3~c |  .0610697
WasDelinq4~c |  .0402655
WasDelinq5~c |  .0269168
-----------------------
```

Borrowers

```
. tabstat CustomerEver* if indivtag, s(mean) c(s)

    variable |      mean
-------------+----------
Customer~1pc |   .753464
Customer~5pc |   .575787
Custome~10pc |  .3832813
Custome~20pc |  .2108416
Custome~30pc |  .1301134
Custome~40pc |  .0868986
Custome~50pc |  .0584606
-----------------------
```

Loan duration by level of delinquency

Source:

```
log:
C:\...\Analyses\Analyses110625_1_DelinqAndRecovery\1106254_0854AB_DelinqAndRecovery.log

use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"
```

Single record model:

```
* Survival time of loans fully repaid.

. * The key idea is to define "fully repaid (no balance)" as the failure event, and last
transaction date as the event time, the loan opening date as the origin.
.
. gen byte NoBalanceAsFail = ( OduePriAmtAtEndOfObs <= 0)

. label var NoBalanceAsFail "No balance open as failure event"


. * One of the problems is that often the delinquency level is reached AFTER the last
cash transaction date. This is the case when there is a balance owed by the borrower at
the end of the observation period:
.
.
. clonevar  OpenBalanceLastObs =  LastCashTransactionDate

. replace  OpenBalanceLastObs =  LastObservationDate if  NoBalanceAsFail == 0

. label var  OpenBalanceLastObs "Date last transact if repaid - else 30 Sept 10".
.
. gsort  IndividualCustomerGId  LoanSeq


. stset   OpenBalanceLastObs , failure(NoBalanceAsFail==1) origin(time OpenDate)

     failure event:  NoBalanceAsFail == 1
obs. time interval:  (origin, OpenBalanceLastObs]
 exit on or before:  failure
    t for analysis:  (time-origin)
            origin:  time OpenDate

--------------------------------------------------------------------------------
   948032  total obs.
     7288  event time missing (OpenBalanceLastObs>=.)           PROBABLE ERROR
        1  obs. end on or before enter()
--------------------------------------------------------------------------------
   940743  obs. remaining, representing
   839791  failures in single record/single failure data
 3.48e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      2299
```

```
. stdes

        failure _d:  NoBalanceAsFail == 1
  analysis time _t:  (OpenBalanceLastObs-origin)
            origin:  time OpenDate

                                |-------------- per subject --------------|
Category                  total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           940743
no. of records            940743           1           1          1          1

(first) entry time                         0           0          0          0
(final) exit time                  369.4346           1        314       2299

subjects with gap              0
time on gap if gap             0
time at risk           3.475e+08    369.4346           1        314       2299

failures                  839791    .8926891           0          1          1
------------------------------------------------------------------------------
```

The survival times are calculated for last loans vs. others, and then also by delinquency level. We exemplify for all loans, for delinquency levels of less than 1, and 1 - 5 percent:

```
. stsum, by(IsLastLoan)

         |                incidence       no. of    |------ Survival time -----|
IsLast~n | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
       0 |    201656121   .0025198        541943        294       317       342
       1 |    145886888   .0022734        398800        196       308       372
---------+--------------------------------------------------------------------
   total |    347543009   .0024164        940743        280       315       350


. * Less than one percent:
. stsum if WasDelinq01pc  == 0, by(IsLastLoan  )

         |                incidence       no. of    |------ Survival time -----|
IsLast~n | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
       0 |     79784873   .0031569        256448        286       305       322
       1 |     27659171   .0054897        154051         82       181       271
---------+--------------------------------------------------------------------
   total |    107444044   .0037574        410499        224       292       315


. * Between 1 percent and 5 percent (0.01 =< x < 0.05):
. stsum if WasDelinq05pc  == 0 & WasDelinq01pc, by(IsLastLoan  )

         |                incidence       no. of    |------ Survival time -----|
IsLast~n | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
       0 |     58514208   .0022075        142023        300       323       354
       1 |     17353329    .002786         56207        246       298       335
---------+--------------------------------------------------------------------
   total |     75867537   .0023398        198230        291       319       349
```

etc. for the other levels of delinquency, as graphed out in the study.

Hazard models

All our survival analysis models of delinquency behavior with covariates used the 10-percent level. These models take place at the borrower level, across all the loans of a borrower.

Main Cox's proportionate hazard model

10-percent delinquency model

```
log:  C:\...\Analyses\Analyses110621_2_MoreCoxModels\110621_1634AB_MoreCoxModels.log
```

```
. use
"C:\...\Analyses\Analyses110608_1_LargeFileEpisodeSplit\110609_1549AB_IndivLoans_10pcDeli
nq_EpisodeSplit_work06.dta"
```

stset done before episode-splitting, result here from split-episode table:

```
. stset
-> stset Delinq10pcOrEnddateAdj, id(IndividualCustomerGId)
                                 failure(failure10pc==2) time0(OpenDate)
                                 exit(time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  failure10pc == 2
obs. time interval:  (OpenDate, Delinq10pcOrEnddateAdj]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate

--------------------------------------------------------------------------------
  3423780  total obs.
     2261  entry on or after exit (OpenDate>Delinq10pcOrEnddateAdj)PROBABLE ERROR
--------------------------------------------------------------------------------
  3421519  obs. remaining, representing
   360492  subjects
   170469  failures in multiple failure-per-subject data
 2.41e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      2300

. stdes

                                  |-------------- per subject --------------|
Category                   total        mean         min     median        max
--------------------------------------------------------------------------------
no. of subjects           360492
no. of records           3421519    9.491248           1          8         37

(first) entry time                   .0570748           0          0        372
(final) exit time                    731.3168           1        608       2300

subjects with gap         206403
time on gap if gap      22670575    56.30203           1         11       2006
time at risk           2.409e+08    668.3719           1        573       2298

failures                  170469    .4728787           0          0          9
--------------------------------------------------------------------------------
```

Covariates

```
. des  splitseq  over*  Over*  PovEntryPoor PovEntryUltra

                 storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------------------------------------
-----------
splitseq        byte   %9.0g                 group(splitcat) [Episodes, usually quarter-years]
over5000        byte   %8.0g                 Loan amount (CPI adj.) over Tk. 5000
over10000       byte   %8.0g                 Loan amount (CPI adj.) over Tk. 10,000
over20000       byte   %8.0g                 Loan amount (CPI adj.) over Tk. 20,000
OverdGr1pc      byte   %8.0g                 Group level overdue preceding quarter-year over 1 percent
OverdGr2pc      byte   %8.0g                 Group level overdue preceding quarter-year over 2 percent
OverdGr4pc      byte   %8.0g                 Group level overdue preceding quarter-year over 4 percent
OverdGr8pc      byte   %8.0g                 Group level overdue preceding quarter-year over 8 percent
OverdMFO1pc     byte   %8.0g                 Organizer level overdue preceding quarter-year over 1 percent
OverdMFO2pc     byte   %8.0g                 Organizer level overdue preceding quarter-year over 2 percent
OverdMFO4pc     byte   %8.0g                 Organizer level overdue preceding quarter-year over 4 percent
OverdBr2pc      byte   %8.0g                 Branch level overdue preceding quarter-year over 2 percent
OverdBr5pc      byte   %8.0g                 Branch level overdue preceding quarter-year over 5 percent
PovEntryPoor    byte   %8.0g                 Implied poverty level at entry: poor
PovEntryUltra   byte   %8.0g                 Implied poverty level at entry: ultra-poor
```

Estimate

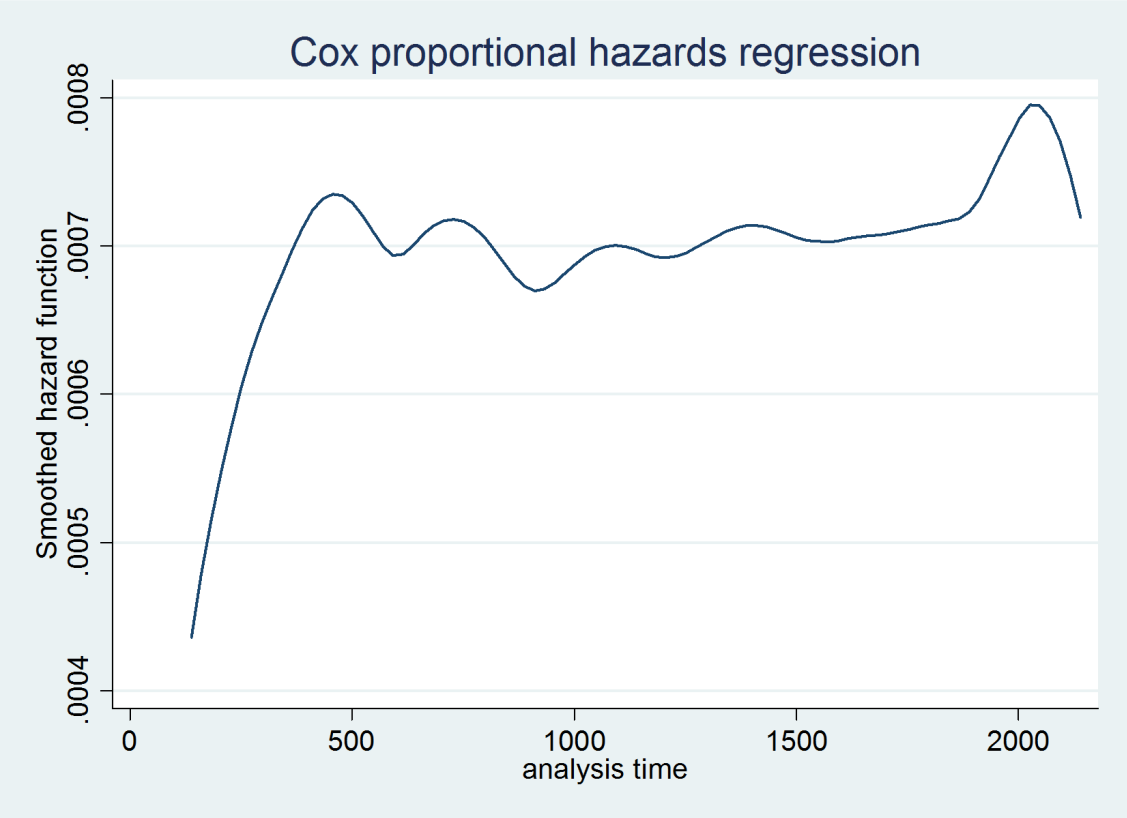
```
. xi: stcox  i.splitseq  over*  Over*  PovEntryPoor PovEntryUltra, basehc(basehc2poverty)
basesurv(basesurv2poverty) nohr
i.splitseq        _Isplitseq_1-25     (naturally coded; _Isplitseq_1 omitted)
```

```
        failure _d:  failure10pc == 2
  analysis time _t:  (Delinq10pcOrEnddateAdj-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId


Cox regression -- Breslow method for ties

No. of subjects =       358489                     Number of obs   =   3312959
No. of failures =       168789
Time at risk    =    233705166
                                                   LR chi2(38)     =  59714.25
Log likelihood  =   -1973406.4                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_Isplitseq_2 |  -.3371084   1.002675    -0.34   0.737    -2.302316    1.628099
_Isplitseq_3 |  -.0910914   1.000786    -0.09   0.927    -2.052595    1.870412
_Isplitseq_4 |  -.4681257   1.000543    -0.47   0.640    -2.429155    1.492903
_Isplitseq_5 |  -.6594053   1.000425    -0.66   0.510    -2.620202    1.301391
_Isplitseq_6 |  -.6887362   1.000331    -0.69   0.491    -2.649349    1.271876
_Isplitseq_7 |  -.6112266   1.000275    -0.61   0.541    -2.571729    1.349276
_Isplitseq_8 |  -.6278212   1.000241    -0.63   0.530    -2.588258    1.332615
_Isplitseq_9 |  -.5320485   1.000214    -0.53   0.595    -2.492432    1.428335
_Isplitse~10 |  -.6013646   1.000203    -0.60   0.548    -2.561726    1.358997
_Isplitse~11 |   -.804253   1.000209    -0.80   0.421    -2.764626     1.15612
_Isplitse~12 |  -.7937735   1.000196    -0.79   0.427    -2.754122    1.166576
_Isplitse~13 |   -.633591   1.000182    -0.63   0.526    -2.593912     1.32673
_Isplitse~14 |  -.6592024   1.000182    -0.66   0.510    -2.619523    1.301118
_Isplitse~15 |   -.659449   1.000181    -0.66   0.510    -2.619768     1.30087
_Isplitse~16 |  -.8238132   1.000184    -0.82   0.410    -2.784137    1.136511
_Isplitse~17 |  -1.114717   1.000194    -1.11   0.265    -3.075061    .8456276
_Isplitse~18 |  -1.042984   1.000188    -1.04   0.297    -3.003315    .9173477
_Isplitse~19 |  -.8680919   1.000178    -0.87   0.385    -2.828405    1.092222
_Isplitse~20 |    -.74862   1.000173    -0.75   0.454    -2.708924    1.211683
_Isplitse~21 |  -.8089061   1.000176    -0.81   0.419    -2.769215    1.151403
_Isplitse~22 |  -.9241393    1.00018    -0.92   0.356    -2.884456    1.036178
_Isplitse~23 |  -1.403682   1.000206    -1.40   0.160    -3.364049    .5566857
_Isplitse~24 |  -1.653183   1.000223    -1.65   0.098    -3.613584    .3072169
_Isplitse~25 |  -1.775564    1.00023    -1.78   0.076     -3.73598    .1848514
    over5000 |   .1162805   .0082447    14.10   0.000     .1001211    .1324398
   over10000 |  -.0947632    .005858   -16.18   0.000    -.1062447   -.0832817
   over20000 |  -.3191396   .0133938   -23.83   0.000    -.3453909   -.2928884
  OverdGr1pc |   .5974041   .0069366    86.12   0.000     .5838087    .6109996
  OverdGr2pc |   .1256042   .0080852    15.54   0.000     .1097575    .1414509
  OverdGr4pc |   .1657993   .0092213    17.98   0.000     .1477259    .1838726
  OverdGr8pc |   .1512321   .0124465    12.15   0.000     .1268374    .1756268
 OverdMFO1pc |   .3777848   .0066966    56.41   0.000     .3646597    .3909099
 OverdMFO2pc |    .070364   .0074576     9.44   0.000     .0557473    .0849807
 OverdMFO4pc |   .0811334   .0104507     7.76   0.000     .0606505    .1016163
OverdBr2pc   |   .0592926   .0058937    10.06   0.000     .0477412    .0708441
OverdBr5pc   |  -.0299428   .0075931    -3.94   0.000    -.0448251   -.0150605
PovEntryPoor |  -.1802061   .0077982   -23.11   0.000    -.1954903   -.1649219
PovEntryUl~a |  -.4484846   .0093588   -47.92   0.000    -.4668275   -.4301418
------------------------------------------------------------------------------
```

Hazard curve

```
. stcurve, hazard outfile(C:\...\Analyses\Analyses110621_2_MoreCoxModels\hazardc2poverty)
```

Figure 3: Main delinquency model - hazard curve



Two additional regional models

Eastern monga (seasonal hunger)-affected area

```
. gen byte MongaArea = (D_ID == 6 | D_ID == 7)

. label var MongaArea "Lalmonirhat and Kurigram Districts"

. tab MongaArea

Lalmonirhat |
        and |
   Kurigram |
  Districts |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |  1,663,378       48.58       48.58
          1 |  1,760,402       51.42      100.00
------------+-----------------------------------
      Total |  3,423,780      100.00
```

Western districts, richer

```
. gen byte ThakPanj = (D_ID == 2 | D_ID ==1)

. tab  ThakPanj

   ThakPanj |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |  2,683,265       78.37       78.37
          1 |    740,515       21.63      100.00
------------+-----------------------------------
      Total |  3,423,780      100.00
```

```
. label var  ThakPanj "Thakurgaon and Panjagarh Districts"
```

Estimates compared

```
. est table _all, stats(N)

----------------------------------------------------
    Variable |   east(monga)   west       allunits
-------------+--------------------------------------
_Isplitseq_2 |  16.707722    9.1175466   -.33710842
_Isplitseq_3 |  17.047172    9.1268544   -.09109142
_Isplitseq_4 |  16.689514    8.6524772   -.46812575
_Isplitseq_5 |  16.376687    8.4757196   -.65940532
_Isplitseq_6 |  16.223284    8.5487528   -.68873621
_Isplitseq_7 |  16.150899    8.6645737   -.61122658
_Isplitseq_8 |  16.209498    8.5408291   -.62782125
_Isplitseq_9 |  16.376351    8.5185373   -.53204849
_Isplitse~10 |  16.208913    8.5323982   -.60136461
_Isplitse~11 |  16.037398    8.3276841     -.804253
_Isplitse~12 |      15.97    8.4780091   -.79377346
_Isplitse~13 |  16.097089    8.6592141   -.63359104
_Isplitse~14 |  15.978957    8.7700672   -.65920238
_Isplitse~15 |  16.120247    8.5633602   -.65944897
_Isplitse~16 |  15.932737    8.4930685   -.82381324
_Isplitse~17 |  15.665154    8.2836759   -1.1147167
_Isplitse~18 |   15.65309     8.559403   -1.0429839
_Isplitse~19 |  15.860779    8.7036605   -.86809186
_Isplitse~20 |  15.773921    8.9742591   -.74862003
_Isplitse~21 |  15.761978    8.8362261   -.80890613
_Isplitse~22 |  15.661665    8.8824222    -.9241393
_Isplitse~23 |  15.314806    8.2824934   -1.4036817
_Isplitse~24 |  15.101874    8.0901495   -1.6531834
_Isplitse~25 |  14.827509    8.1225011   -1.7755642
    over5000 |  .05820849    .16353549    .11628046
   over10000 | -.06580833   -.08576888   -.09476319
   over20000 | -.38747109   -.21659461   -.31913963
  OverdGr1pc |  .66168066    .38884479    .59740413
  OverdGr2pc |  .16044763    .09025419     .1256042
  OverdGr4pc |  .21116203     .1453217    .16579929
  OverdGr8pc |  .17934657    .12801027    .15123213
 OverdMFO1pc |  .35454334     .3516695     .3777848
 OverdMFO2pc |  .04811726    .04408637    .07036403
 OverdMFO4pc |  .11737217    .10997023    .08113344
OverdBr2_5pc |   .1276682    .07537751    .05929264
OverdBr5_0pc | -.05773104    .18432434   -.02994285
PovEntryPoor |  -.2937298    -.0775247   -.18020608
PovEntryUl~a | -.42407737   -.32440736   -.44848462
-------------+--------------------------------------
           N |    1672042       732709      3312959
----------------------------------------------------
```

Quarter-year coefficients are not directly comparable. They were subsequently centered within each model and compared in a period graph; see study report:

```
.                                                                                                  use
C:\...\Analyses\Analyses110621_2_MoreCoxModels\110621_2018AB_QuarterYearEffects_2Regions.
dta


. des  QuarterYearNumber st_monga st_west st_allunits

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------------
--------------------------------
QuarterYearNu~r int    %tq                    Quarter-year
st_monga        float  %9.0g                  Lalmonirhat and Kurigram
st_west         float  %9.0g                  Thakurgaon and Panchagarh
st_allunits     float  %9.0g                  All program units

. twoway (line   st_allunits QuarterYearNumber, sort lwidth(thick))   (line    st_monga
QuarterYearNumber,  sort  lwidth(medium))    (line      st_west  QuarterYearNumber,  sort
lwidth(medium)), ytitle(Standardized regression coefficients) yline(0,  lwidth(medthick)
lcolor(gs12))  title(Period  effects  on  loan  delinquency)  note("Note:  Cox  proportional
hazard  model  of  time  to  delinquency  (10  percent  of  the  principal).""Controlling  for
organizational, borrower and loan factors. 360,492 borrowers in the entire ""program area.
Coeff. standard.: centered with mean = 0 and SD = 1.")
```

Borrower-level analyses

In and outflow of borrowers

Source:

```
.    use    "C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep2010_tagsCORR_work07.dta"
```

Life table approach

Entries

```
. * Using ltable with dead = 0 to get counts of new entries (as Lost , paradoxically) in interval.
. * Half-yearly intervals, starting with 16233 = 1 June 2004 in STATA, then from 1 Jul 2004
. * Then as num list; 16233 16253 16437 16618 16802 16983 17167 17348 17532 17714 17898 18079 18263 18444 18535

. gsort  IndividualCustomerGId  LoanSeq

. gen byte dead = 0

. count if IsFirstLoan
405162

. ltable   OpenDate   dead if IsFirstLoan, i(16233 16253 16437 16618 16802 16983 17167 17348 17532 17714 17898 18079 18263 18444 18535) f
```

Interval	Beg. Total	Deaths	Lost	Cum. Failure	Std. Error	[95% Conf.	Int.]
16233 16253	405162	0	923	0.0000	0.0000	.	.
16253 16437	404239	0	13717	0.0000	0.0000	.	.
16437 16618	390522	0	30474	0.0000	0.0000	.	.
16618 16802	360048	0	40444	0.0000	0.0000	.	.
16802 16983	319604	0	41020	0.0000	0.0000	.	.
16983 17167	278584	0	39760	0.0000	0.0000	.	.
17167 17348	238824	0	32888	0.0000	0.0000	.	.
17348 17532	205936	0	21251	0.0000	0.0000	.	.
17532 17714	184685	0	46395	0.0000	0.0000	.	.
17714 17898	138290	0	35876	0.0000	0.0000	.	.
17898 18079	102414	0	35041	0.0000	0.0000	.	.
18079 18263	67373	0	22705	0.0000	0.0000	.	.
18263 18444	44668	0	29352	0.0000	0.0000	.	.
18444 18535	15316	0	15205	0.0000	0.0000	.	.
18535 .	111	0	111	0.0000	0.0000	.	.

Exits

```
. * Exits:  Defined as last loans, last cash transaction more than three months back from end of observation period:

. count if IsLastLoan & mdy(9,30,2010) -  LastCashTrDateByBorrower > 90
211899

. replace dead = 1 if IsLastLoan & mdy(9,30,2010) -  LastCashTrDateByBorrower > 90

. ltable   OpenDate   dead if   IsLastLoan, i(16233 16253 16437 16618 16802 16983 17167 17348 17532 17714 17898 18079 18263 18444 18535) f
```

Interval	Beg. Total	Deaths	Lost	Cum. Failure	Std. Error	[95% Conf.	Int.]
16233 16253	405162	214	0	0.0005	0.0000	0.0005	0.0006
16253 16437	404948	3434	9	0.0090	0.0001	0.0087	0.0093
16437 16618	401505	9139	28	0.0316	0.0003	0.0310	0.0321
16618 16802	392338	16462	95	0.0722	0.0004	0.0714	0.0730
16802 16983	375781	18929	160	0.1189	0.0005	0.1180	0.1199
16983 17167	356692	23466	298	0.1769	0.0006	0.1758	0.1781
17167 17348	332928	21216	420	0.2294	0.0007	0.2281	0.2307
17348 17532	311292	20462	488	0.2801	0.0007	0.2787	0.2815
17532 17714	290342	27870	704	0.3493	0.0008	0.3478	0.3508
17714 17898	261768	28665	1461	0.4208	0.0008	0.4192	0.4223

```
17898 18079    231642    32951   3332     0.5037    0.0008     0.5022    0.5053
18079 18263    195359     6424  37127     0.5218    0.0008     0.5202    0.5233
18263 18444    151808      801 100459     0.5255    0.0008     0.5240    0.5271
18444 18535     50548     1729  48480     0.5567    0.0010     0.5547    0.5588
18535     .       339      137    202     0.8119    0.0142     0.7833    0.8389
-------------------------------------------------------------------------------

. * Results transferred to Excel; Graph made in Excel.

. drop dead /// * No irony intended.
```

Further processing in Excel

C:\...\Analyses\Analyses110619_2_BorrowerFlow\110619_1549AB_BorrowerFlowDiagram.xlsx

Period EndDate	STATA daynumbering	New entrants	Exits during period	Ending period	Previous period
01-Jul-04	16253	923	214	709	0
01-Jan-05	16437	13717	3434	10,992	709
01-Jul-05	16618	30474	9139	32,327	10,992
01-Jan-06	16802	40444	16462	56,309	32,327
01-Jul-06	16983	41020	18929	78,400	56,309
01-Jan-07	17167	39760	23466	94,694	78,400
01-Jul-07	17348	32888	21216	106,366	94,694
01-Jan-08	17532	21251	20462	107,155	106,366
01-Jul-08	17714	46395	27870	125,680	107,155
01-Jan-09	17898	35876	28665	132,891	125,680
01-Jul-09	18079	35041	32951	134,981	132,891
01-Jan-10	18263	22705	6424	151,262	134,981
01-Jul-10	18444	29352	801	179,813	151,262
30-Sep-10	18535	15205	1729	193,289	179,813

See graph based on table, in study report.

Length of partnership

Observed length

We define the *observed* length of partnership as time elapsed between the opening of the borrower's first loan and the earlier of repayment or last transaction across any loans (usually the last). Obviously, this is not the same as the estimated duration including for those borrowers who had an open loan running normally at the end of the global observation period.

We corrected the calculation of this variable late in the analysis process. We had earlier calculated to the repayment of the last loan. However, what is needed is the minimum, for each loan of a given borrower, of loan repayment or last cast transaction dates, and then the maximum for these.

```
. egen EarlierOfRepayOrLastTr = rowmin(  LastCashTransactionDate  LoanRepaidDate)

.  label  var    EarlierOfRepayOrLastTr  "Earlier  of  last  loan  repayment  or  last  cash
transaction date"
```

```
.      by      IndividualCustomerGId:      egen      MaxAllLoansMinLastRepayOrTr      =
max( EarlierOfRepayOrLastTr)

. clonevar EndObservedPartnerDate =  MaxAllLoansMinLastRepayOrTr

. di mdy(9,30,2010)
18535

. replace  EndObservedPartnerDate = 18535 if  EndObservedPartnerDate == .
(1495 real changes made)

. gen LengthPartner =  EndObservedPartnerDate -  DateFirstLoanOpened
(584532 real changes made)

label var LengthPartner "Length of partnership (in days)"

. summ LengthPartner, detail

              Length of partnership (in days)
-------------------------------------------------------------
      Percentiles      Smallest
 1%           47              0
 5%          248              0
10%          315              0       Obs              948032
25%          604              0       Sum of Wgt.      948032

50%          965                      Mean           1039.619
                      Largest       Std. Dev.      561.2112
75%         1484          2300
90%         1841          2300       Variance         314958
95%         2000          2300       Skewness        .2178662
99%         2179          2300       Kurtosis        2.021874


. gen  LengthPartnerYears = floor( LengthPartner / 365.242199)

. label var LengthPartnerYears "Length of partnership (in years, rounded down)"
```

Expected length, survival analysis

Source:

```
log:
C:\...\Analyses\Analyses110705_1_LengthPartnershipCORR\1107050_1655AB_LengthPartnerByYear
FirstLoanCORR.log

.                                                                                    use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------
LastCashTrans~e int    %tdD_m_Y               Date of last cash transaction
IndividualC~GId long   %12.0g                 Customer ID (unique to this dataset)
LengthPartner~t byte   %8.0g                  = 1 if last loan repaid or, else, 30 Sep 10
                                              - date last cash transact > 90 days; else 0
OpenDate        int    %tdD_m_Y               Date loan disbursed


.      stset           LastCashTransactionDate           ,      id(IndividualCustomerGId)
failure(LengthPartnerFailureEvent==1) time0(OpenDate) exit(
> time .) origin(time OpenDate)

                id:  IndividualCustomerGId
     failure event:  LengthPartnerFailureEvent == 1
obs. time interval:  (OpenDate, LastCashTransactionDate]
 exit on or before:  time .
    t for analysis:  (time-origin)
            origin:  time OpenDate

--------------------------------------------------------------------------------
   948032  total obs.
     7445  event time missing (LastCashTransactionDate>=.)      PROBABLE ERROR
        1  entry on or after exit (OpenDate>LastCashTransactionDate)PROBABLE ERROR
    79443  overlapping records (LastCashTransactionDate[_n-1]>OpenDate)PROBABLE ERROR
--------------------------------------------------------------------------------
   861143  obs. remaining, representing
   403627  subjects
```

```
   208719  failures in single failure-per-subject data
 2.66e+08  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      2300

. stdes

                                 |-------------- per subject --------------|
Category                  total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           403627
no. of records            861143    2.133512           1          2         12

(first) entry time                  5.005111           0          0       1694
(final) exit time                   724.9087           1        613       2300

subjects with gap         226416
time on gap if gap      24685955    55.51185           1         10       1722
time at risk           2.659e+08    658.7433           1        580       2281

failures                  208719    .5171086           0          1          1
------------------------------------------------------------------------------


. stsum

         |               incidence       no. of    |------ Survival time -----|
YearFi~d | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
   total |    265886562    .000785        403627        364       788      1494

. di 1 / 0.000785
1273.8854 days [mean] vs. 788 days [median]
```

Expected length, by year of first loan opened

```
. stsum, by( YearFirstLoanOpened)

        failure _d:  LengthPartnerFailureEvent == 1
  analysis time _t:  (LastCashTransactionDate-origin)
            origin:  time OpenDate
 exit on or before:  time .
                id:  IndividualCustomerGId

         |               incidence       no. of    |------ Survival time -----|
YearFi~d | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
    2004 |     15051585   .0006857         14623        495       966      1946
    2005 |     65107318   .0007781         70859        367       907      1668
    2006 |     67691076   .0008085         80754        366       910      1496
    2007 |     40087767   .0008417         54135        390       820      1343
    2008 |     48768382   .0008519         82266        363       700         .
    2009 |     23407354   .0007501         57729        345       633         .
    2010 |      5773080   .0000293         43261          .         .         .
---------+--------------------------------------------------------------------
   total |    265886562    .000785        403627        364       788      1494
```

The incidence rates for 2004 - 2009 were the basis for the corresponding expected mean length of partnership in the executive summary of the study report.

Dynamic incentives

We investigated the incidence of loans after the first, the growth of principal from first to second over time, and the effects of loan and borrower covariates on the probability and amount of second loans. We do not report the simpler types of descriptive statistics here.

Wide-shape file creation

Using STATA's *reshape* command, we created a wide-shape file for a small number of variables so that second loan variables could be related to first loan and borrower characteristics. From

```
log:
C:\...\Analyses\Analyses110620_3_LengthPartnership\110620_2155AB_SubsequentLoand.log

C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep2010
_tagsCORR_work07.dta"

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------
--------------------------------
U_id            byte   %8.0g                  RDRS Unit identifier
GroupGId        int    %12.0g                 Group GId
DisbursedAmt    float  %9.0g                  Loan amount disbursed
DisbursedAmtC~j float  %9.0g                  Loan principal (adj. to CPI Sep 2010)
logPrincipCPI    float  %9.0g                     Loan principal (log10) - adj to CPI Sep
2010
OpenDate        int    %tdD_m_Y               Date loan disbursed
WasDelinq01pc   byte   %8.0g                  Loan was delinquent > 01pc of principal
WasDelinq05pc   byte   %8.0g                  Loan was delinquent > 05pc of principal
WasDelinq10pc   byte   %8.0g                  Loan was delinquent > 10pc of principal
WasDelinq20pc   byte   %8.0g                  Loan was delinquent > 20pc of principal
WasDelinq30pc   byte   %8.0g                  Loan was delinquent > 30pc of principal
WasDelinq40pc   byte   %8.0g                  Loan was delinquent > 40pc of principal
WasDelinq50pc   byte   %8.0g                  Loan was delinquent > 50pc of principal
IndividualC~GId long   %12.0g                 Customer ID (unique to this dataset)
LoanSeq         byte   %9.0g                  Sequence of loan for individual customer
GroupIsFemale    byte    %8.0g                     Customer is member of women's borrower
group
PovEntryPoor    byte   %8.0g                  Implied poverty level at entry: poor
PovEntryUltra   byte   %8.0g                  Implied poverty level at entry: ultra-poor
CVLoansFirstY~p float  %9.0g                    Coeff. var. loan size within group in its
first year
NoLoansFirstY~p float  %9.0g                    Number loans taken by group in its first
year

.                                                                                   save
"C:\...\Analyses\Analyses110620_4_SubsequentLoans\110620_2157AB_SubsequentLoans_LongShape
.dta"

. keep if LoanSeq < 5

. keep U_id  GroupGId DisbursedAmt DisbursedAmtCPIadj logPrincipCPI OpenDate WasDelinq*pc
IndividualCustomerGId    LoanSeq       GroupIsFemale       PovEntryPoor    PovEntryUltra
CVLoansFirstYearOfGroup NoLoansFirstYearOfGroup

.
. reshape wide   U_id   GroupGId DisbursedAmt DisbursedAmtCPIadj logPrincipCPI OpenDate
WasDelinq*pc   GroupIsFemale   PovEntryPoor   PovEntryUltra      CVLoansFirstYearOfGroup
NoLoansFirstYearOfGroup, i( IndividualCustomerGId) j( LoanSeq)
(note: j = 1 2 3 4)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                   874542   ->  405162
Number of variables                  20   ->      73
j variable (4 values)           LoanSeq   ->   (dropped)
xij variables:
                                    U_id   ->   U_id1 U_id2 ... U_id4
                                GroupGId   ->   GroupGId1 GroupGId2 ... GroupGId4
                            DisbursedAmt   ->   DisbursedAmt1 DisbursedAmt2 ... DisbursedAmt4
                     DisbursedAmtCPIadj   ->   DisbursedAmtCPIadj1 DisbursedAmtCPIadj2 ... DisbursedAmtCPIadj4
                          logPrincipCPI   ->   logPrincipCPI1 logPrincipCPI2 ... logPrincipCPI4
                              OpenDate   ->   OpenDate1 OpenDate2 ... OpenDate4
                          WasDelinq01pc   ->   WasDelinq01pc1 WasDelinq01pc2 ... WasDelinq01pc4
                          WasDelinq05pc   ->   WasDelinq05pc1 WasDelinq05pc2 ... WasDelinq05pc4
                          WasDelinq10pc   ->   WasDelinq10pc1 WasDelinq10pc2 ... WasDelinq10pc4
                          WasDelinq20pc   ->   WasDelinq20pc1 WasDelinq20pc2 ... WasDelinq20pc4
                          WasDelinq30pc   ->   WasDelinq30pc1 WasDelinq30pc2 ... WasDelinq30pc4
                          WasDelinq40pc   ->   WasDelinq40pc1 WasDelinq40pc2 ... WasDelinq40pc4
                          WasDelinq50pc   ->   WasDelinq50pc1 WasDelinq50pc2 ... WasDelinq50pc4
                          GroupIsFemale   ->   GroupIsFemale1 GroupIsFemale2 ... GroupIsFemale4
                           PovEntryPoor   ->   PovEntryPoor1 PovEntryPoor2 ... PovEntryPoor4
                          PovEntryUltra   ->   PovEntryUltra1 PovEntryUltra2 ... PovEntryUltra4
               CVLoansFirstYearOfGroup   ->   CVLoansFirstYearOfGroup1 CVLoansFirstYearOfGroup2 ...
                                                     CVLoansFirstYearOfGroup4
               NoLoansFirstYearOfGroup   ->   NoLoansFirstYearOfGroup1 NoLoansFirstYearOfGroup2 ...
                                                     NoLoansFirstYearOfGroup4
-----------------------------------------------------------------------------
```

Growth rate

Loan size (CPI-adjusted) for first four loans:

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
Disbursed~j1 |    405162    6663.523    3579.088   73.15757   136031.8
Disbursed~j2 |    245148    9076.675    4563.692   52.07607   151713.4
Disbursed~j3 |    144173    11115.77    6583.742   108.5284   227140.7
Disbursed~j4 |     80059    13283.46    9414.182   102.1096   322905.9


. * Growth rates:
. * From first to second:
.
. gen GrowthLog1_2 =  logPrincipCPI2 - logPrincipCPI1

. label var  GrowthLog1_2 "Change in loan amount from 1st to 2nd loan (log10)"

. summ  GrowthLog1_2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
GrowthLog1_2 |    245148    .1400093    .1414658  -2.329262          2
```

Dynamic incentives graph

Growth rate from first to second loan, over time, using STATA's local polynomial smoothing command *lpoly:*

```
. display mdy(7,1,2005)
16618

. lpoly   GrowthRate1_2 OpenDate2, gen(r1x r1s) se(ser1s) nograph bw(100) n(100)

. format  r1x  %tdD_m_Y

. twoway (line r1s r1x, sort lwidth(medthick)) if r1x > 16617 ytitle(Ratio of loan
principals) yline(0, lpattern(dash)) xtitle(Date second loan was issued) title(Dynamic
incentives over time) subtitle(Ratio of second loan size to first loan size) note("Note:
405,162 borrowers with a first loan, 245,148 with a second loan. Inflation adjusted.
Smoothed means.""Plot restricted to second loans issued after 30 June 2005.", span)
```

See chart in study report.

Tobit model of obtaining a second loan and of its size

Using the alternative *craggit* model (Burke 2009).

```
C:\...\Analyses\Analyses110620_4_SubsequentLoans\110621_1100AB_CraggitSecondLoan.log

. gen   logPrincipCPI2Zero = logPrincipCPI2

. replace  logPrincipCPI2Zero = 0 if logPrincipCPI2 ==.
. * [Because of the Tobit construction]
```

Estimates

```
.  craggit       logPrincipCPI2Zero  logPrincipCPI1     WasDelinq10pc1     WasDelinq30pc1
WasDelinq50pc1      NoLoansFirstYearOfGroup1   CVLoansFirstYearOfGroup1      PovEntryPoor1
PovEntryUltra1   if   OpenDate1   <   18171   &   NoLoansFirstYearOfGroup1   <   11514,
second(logPrincipCPI2Zero logPrincipCPI1   WasDelinq10pc1   WasDelinq30pc1 WasDelinq50pc1
NoLoansFirstYearOfGroup1   CVLoansFirstYearOfGroup1       PovEntryPoor1   PovEntryUltra1)
vce(robust)

Estimating Cragg's tobit alternative
Assumes conditional independence

                                                Number of obs   =     301812
                                                Wald chi2(8)    =   56584.94
Log pseudolikelihood =  8223.4184               Prob > chi2     =     0.0000

------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
```

```
-------------+----------------------------------------------------------------
Tier1        |
logPrincip~1 |   .4291228   .0169224    25.36   0.000     .3959555    .4622902
WasDelinq1~1 |  -1.305802   .0071797  -181.87   0.000    -1.319874    -1.29173
WasDelinq3~1 |  -.4465528   .0146155   -30.55   0.000    -.4751987    -.417907
WasDelinq5~1 |  -.1307082   .0204034    -6.41   0.000    -.1706982   -.0907182
NoLoansFir~1 |   .0076523   .0002347    32.61   0.000     .0071924    .0081123
CVLoansFir~1 |   .6242047   .0195739    31.89   0.000     .5858406    .6625688
PovEntryPo~1 |   .0206572   .0096544     2.14   0.032      .001735    .0395794
PovEntryUl~1 |  -.1018145   .0111042    -9.17   0.000    -.1235783   -.0800508
       _cons |  -.9882189   .0663522   -14.89   0.000    -1.118267    -.858171
-------------+----------------------------------------------------------------
Tier2        |
logPrincip~1 |   .6558784   .0021332   307.46   0.000     .6516974    .6600594
WasDelinq1~1 |  -.0709841   .0012916   -54.96   0.000    -.0735155   -.0684526
WasDelinq3~1 |  -.0455117   .0034191   -13.31   0.000    -.0522131   -.0388104
WasDelinq5~1 |   .0517523   .0051117    10.12   0.000     .0417336     .061771
NoLoansFir~1 |   .0003529   .0000232    15.23   0.000     .0003075    .0003983
CVLoansFir~1 |   .0369021   .0023158    15.93   0.000     .0323632     .041441
PovEntryPo~1 |  -.0041962   .0009554    -4.39   0.000    -.0060688   -.0023237
PovEntryUl~1 |  -.0232507   .0011039   -21.06   0.000    -.0254144    -.021087
       _cons |   1.434729   .0083618   171.58   0.000     1.418341    1.451118
-------------+----------------------------------------------------------------
sigma        |
       _cons |   .1233725   .0005045   244.54   0.000     .1223837    .1243613
------------------------------------------------------------------------------
. * Note: NoLoansFirstYearOfGroup1 < 11514, because this is the number of first loans
(i.e. customers) for which the loan table did not have a group ID. They were lumped
together in the calculation of this variable.
```

Predictions

```
. * About predict in craggit, see Stata Journal 9-4 p. 589
.
. predict x1g2 if e(sample), eq(Tier1)

. predict x2g2 if e(sample), eq(Tier2)

. predict sigma2 if e(sample), eq(sigma)

. gen ProbGets2ndLoan2 = normal(x1g2)

.
. summ  x1g-ProbGets2ndLoan2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        x1g2 |    301812    .7134158      .61198  -1.674592   2.249373
        x2g2 |    301812    3.908956    .1274421   2.595351   4.788962
      sigma2 |    301812    .1233725           0   .1233725   .1233725
ProbGets2n~2 |    301812    .7384996    .2103453   .0470072   .9877556

. summ logPrincipCPI2 if OpenDate1 < 18171

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
logPrincip~2 |    238858    3.919176    .1767794   1.724763   5.181024

Colors indicate corresponding observed and predicted variables.
```

Default

Default models were estimated, assuming that most defaulted-on loans were last loans in the borrowers' careers with RDRS. Because we consider last loans only, these models are borrower-level models.

Source:

```
log:  C:\...\Analyses\Analyses100625_2_Default\1106254_1131AB_Default.log

and

C:\...\Analyses\Analyses110626_1_DefaultCORR\110626_0914AB_DefaultCraggitCORR.log

.                                                                              use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"
```

Incidence

```
. des DisbursedAmt BalanceOnMaturityDate OduePriAmtAtEndOfObs NoBalanceAsFail

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------
DisbursedAmt    float  %9.0g                  Loan amount disbursed
BalanceOnMatu~e float  %9.0g                  Loan balance on maturity
OduePriAmtAtE~s float  %9.0g                  Overdue on principal by end of observation
                                              period
NoBalanceAsFail byte   %8.0g                  No balance open as failure event

. summ DisbursedAmt BalanceOnMaturityDate OduePriAmtAtEndOfObs NoBalanceAsFail

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
DisbursedAmt |    948032    7926.089    6466.318         50     400000
BalanceOnM~e |    743547    334.4312    1310.067  -26407.48      82500
OduePriAmt~s |    948032    66.90275    946.8122     -50000      74750
NoBalanceA~l |    948032    .8935141    .3084586          0          1

. * Limited to those last loans for which no cash transaction took place in the 3 months
prior to 30 Sept 201:

. tab IsLastLoan  NoBalanceAsFail if   mdy(9, 30,2010) -  LastCashTransactionDate > 90 ,
V

 Last loan |
        in |  No balance open as
  customer |     failure event
   history |         0          1 |     Total
-----------+----------------------+----------
         0 |    29,783    477,583 |   507,366
         1 |    45,861    170,569 |   216,430
-----------+----------------------+----------
     Total |    75,644    648,152 |   723,796

              Cramér's V =  -0.2293
```

Incidence rate from survival model

```
log:
C:\...\Analyses\Analyses110626_1_DefaultCORR\110626_1426AB_CalculationOfDefaultRate.log

.                                                                                   use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"


. stset  LastCashTransactionDateAdj if include &  mdy(9, 30, 2010) - MaturityDate >= 0 &
mdy(9,      30,      2010)      -            LastCashTransactionDateAdj      >      90,
failure(  OdueofDisbursedAbove10pc==1) id( IndividualCustomerGId) origin(time OpenDate)

["gen byte OdueofDisbursedAbove10pc", see further below. Here use as failure variable.]

                id:  IndividualCustomerGId
     failure event:  OdueofDisbursedAbove10pc == 1
obs. time interval:  (LastCashTransactionDateAdj[_n-1], LastCashTransactionDateAdj]
 exit on or before:  failure
    t for analysis:  (time-origin)
            origin:  time OpenDate
            if exp:  include &  mdy(9, 30, 2010) - MaturityDate >= 0 & mdy(9, 30, 2010) -
                      LastCashTransactionDateAdj > 90

------------------------------------------------------------------------------
   948032  total obs.
   293414  ignored at outset because of -if <exp>-
      173  event time missing (LastCashTransactionDateAdj>=.)   PROBABLE ERROR
      424  multiple records at same instant                     PROBABLE ERROR
           (LastCashTransactionDateAdj[_n-1]==LastCashTransactionDateAdj)
      705  obs. end on or before enter()
     3219  obs. begin on or after (first) failure
------------------------------------------------------------------------------
   650097  obs. remaining, representing
   297619  subjects
    31840  failures in single failure-per-subject data
 2.26e+08  total analysis time at risk, at risk from t =         0
                              earliest observed entry t =         0
                                   last observed exit t =      2207
```

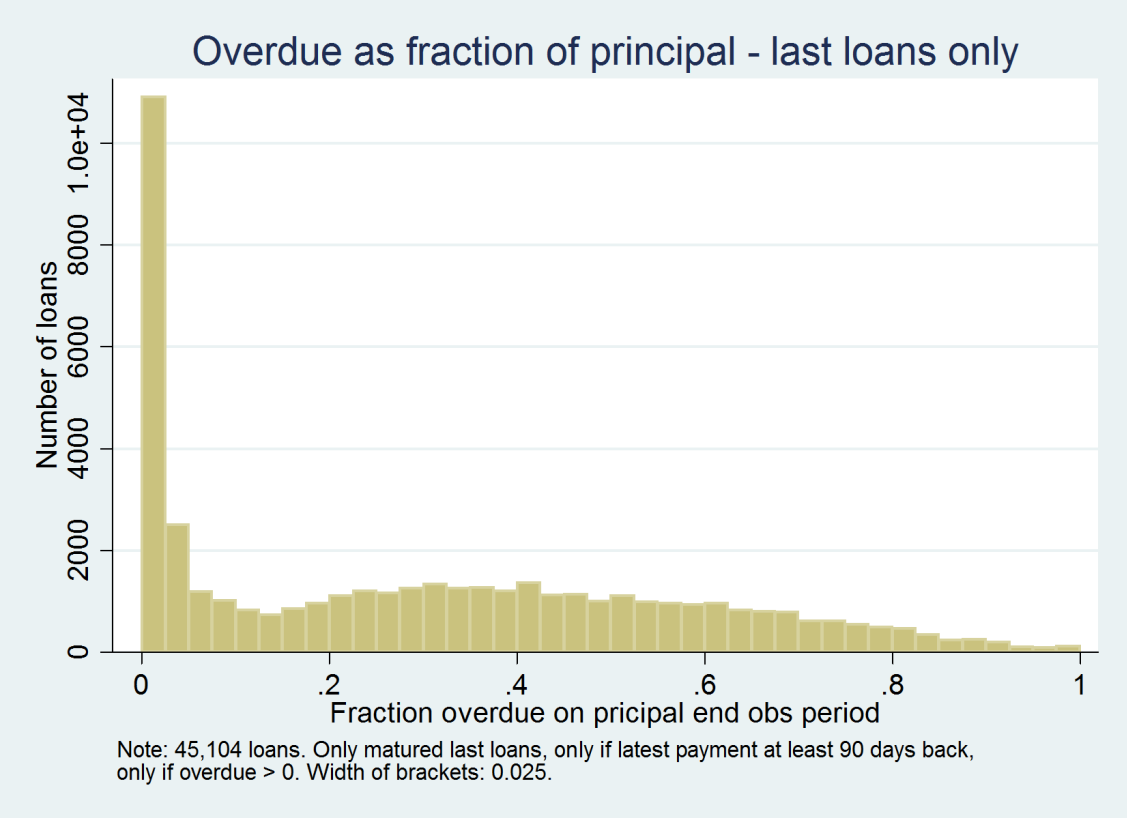
```
. stdes

                                |-------------- per subject --------------|
Category                  total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           297619
no. of records            650097    2.184326           1          2         13

(first) entry time                  .0487469           0          0        394
(final) exit time                   760.8752           1        648       2207

subjects with gap             54
time on gap if gap         14231     263.537           7        316        666
time at risk           2.264e+08    760.7786           1        648       2207

failures                   31840    .1069824           0          0          1
------------------------------------------------------------------------------

. stsum

         |              incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
   total |    226422172   .0001406        297619       1503      2052         .


. * Days per incident:
. di 1/ .0001406
7112.3755 days

. * Years per incident:
. di 7112.3755 / 365.24
19.473156 years
```

As a fraction of the principal

```
. gen OdueLog10 = log10( OduePriAmtAtEndOfObs)

. label var  OdueLog10 "Overdue on principal end obs period - log10"

. summ  OdueLog10

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
   OdueLog10 |    100952    2.295015    1.360838         -2   4.873611

. replace  OdueLog10 = 0 if  OdueLog10 ==.

. replace  OdueLog10 = 0 if  OdueLog10 < 0

. replace  OdueLog10 = . if MaturityDate > mdy(9,30,2010)


. count if IsLastLoan & mdy(9, 30,2010) - MaturityDate >= 0 & mdy(9, 30,2010) -
LastCashTransactionDate > 90
208988

. gen Odue100930ofDisbursed = OduePriAmtAtEndOfObs / DisbursedAmt

. label var Odue100930ofDisbursed "Fraction overdue on pricipal end obs period"

Fraction overdue on pricipal end obs period

. summ Odue100930ofDisbursed

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
Odue100930~d |    948032    .0106357    .1010508  -.9998633          1

. summ Odue100930ofDisbursed if IsLastLoan

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
Odue100930~d |    405162    .0226321    .1499083  -.9998633          1
```

```
. histogram Odue100930ofDisbursed if IsLastLoan &  Odue100930ofDisbursed > 0  & mdy(9, 30,2010) - MaturityDate >= 0 & mdy(9, 30,2010) - LastCashTransactionDate > 90, freq width(0.025) start(0) note("Note: 45,104 loans. Only matured last loans, only if latest payment at least 90 days back," "only if overdue > 0. Width of brackets: 0.025.") title(Overdue as fraction of principal - last loans only) ytitle(Number of loans)
```

Figure 4: Overdue as a fraction of the loan principal - last loans only



Needed for the Tobit model further below:

```
. gen byte  OdueofDisbursedAbove10pc = ( Odue100930ofDisbursed > 0.1)

. label var   OdueofDisbursedAbove10pc "Overdue on 30 Sept 2010 higher than 10 pc principal"

. gen OdueOfDisbursedIfAbove10pc = Odue100930ofDisbursed
. replace OdueOfDisbursedIfAbove10pc = 0 if Odue100930ofDisbursed <= 0.1
[= 0, not = . because of the Tobit character]

. label var  OdueOfDisbursedIfAbove10pc "Takes default rate if default > 10 pc - else 0 [Craggit model]"
```

As a fraction of total loan sum

```
. * Display sum of principals and of overdue in matured loans, with last payment more than 90 days back:

. table IsLastLoan  NoBalanceAsFail if include &  mdy(9, 30,2010) - MaturityDate >= 0 & mdy(9, 30,2010) - LastCashTransactionDateAdj > 90 , c(freq sum   DisbursedAmt  sum OduePriAmtAtEndOfObs) row col

----------------------------------------------
Last loan |
in        |
customer  |No balance open as failure event
history   |         0          1      Total
```

```
----------+---------------------------------
        0 |    31,268    458,611    489,879
          |  2.48e+08   3.40e+09   3.65e+09
          |  1.03e+07  -32686.23   1.02e+07
          |
        1 |    44,978    119,761    164,739
          |  3.22e+08   8.27e+08   1.15e+09
          |  1.00e+08  -14108.81   1.00e+08
          |
    Total |    76,246    578,372    654,618
          |  5.70e+08   4.23e+09   4.80e+09
          |  1.11e+08  -46795.04   1.11e+08
-------------------------------------------

.
. * Ratio of overdue (overdue from No balance = 0) and principal [both from row Total,
last loan irrelevant here except for control purposes]:
.
end of do-file

. di 1.11 / 48
.023125,  i.e.  2.3  percent  of  the  cumulative  matured  loan  sum  is  estimated  to  be  in
default.
```

Tobit model with covariates

We estimate alternative Tobit models (STATA's *craggit*) (Burke 2009) for the fraction of the principal defaulted on. Given the above distribution, which likely mixed two or more unknown processes, we limit cases to fractions above 10 percent. We document the estimate of the model including imputed poverty levels at time of program entry. Another model, instead using the gender of the borrower, had very similar results.

Variables

```
. des  OdueOfDisbursedIfAbove10pc   WasDelinq*pcPreviousLoan  Overd*  over*  Year*  PovEntryPoor  PovEntryUltra
include  IsLastLoan MaturityDate LastCashTransactionDate OpenDate

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------------------------
OdueOfDisburs~c float  %9.0g                 Takes default rate if default > 10 pc - else 0 [Craggit model]
WasDelinq10pc~n byte   %8.0g                 Borrower was previously delinquent at 10 percent level
WasDelinq30pc~n byte   %8.0g                 Borrower was previously delinquent at 30 percent level
WasDelinq50pc~n byte   %8.0g                 Borrower was previously delinquent at 50 percent level
over5000        byte   %8.0g                 Loan amount (CPI adj.) over Tk. 5000
over10000       byte   %8.0g                 Loan amount (CPI adj.) over Tk. 10,000
over20000       byte   %8.0g                 Loan amount (CPI adj.) over Tk. 20,000
OverdGr1pcOD    byte   %8.0g                 Group level overdue preceding loan disbursement over 1 percent
OverdGr2pcOD    byte   %8.0g                 Group level overdue preceding loan disbursement over 2 percent
OverdGr4pcOD    byte   %8.0g                 Group level overdue preceding loan disbursement over 4 percent
OverdGr8pcOD    byte   %8.0g                 Group level overdue preceding loan disbursement over 8 percent
OverdMFO1pcOD   byte   %8.0g                 Organizer level overdue preceding loan disb. over 1 percent
OverdMFO2pcOD   byte   %8.0g                 Organizer level overdue preceding loan disb. over 2 percent
OverdMFO4pcOD   byte   %8.0g                 Organizer level overdue preceding loan disb. over 4 percent
OverdBr2pcOD    byte   %8.0g                 Branch level overdue preceding loan disbursement over 2 percent
OverdBr5pcOD    byte   %8.0g                 Branch level overdue preceding loan disbursement over 5 percent
YearFirstLoan~d float  %9.0g                 Year first loan was opened
Year2006        byte   %8.0g                 First loan was opened in 2006
Year2007        byte   %8.0g                 First loan was opened in 2007
Year2008        byte   %8.0g                 First loan was opened in 2008
Year2009or10    byte   %8.0g                 First loan was opened in 2009 or 2010
PovEntryPoor    byte   %8.0g                 Implied poverty level at entry: poor
PovEntryUltra   byte   %8.0g                 Implied poverty level at entry: ultra-poor
include         byte   %8.0g                 Included if MFOrganizerGid and GroupGId both present
IsLastLoan      byte   %8.0g                 Last loan in customer history
MaturityDate    int    %tdD_m_Y              Date loan matured
LastCashTrans~e int    %tdD_m_Y              Date of last cash transaction
OpenDate        int    %tdD_m_Y              Date loan disbursed
```

Estimate

In retrospect, the motivation to exclude last loans opened between 13 June 2004 and 30 June 2005 (see command below) is not remembered. We may have considered the high probability that someone taking her last loan during the first year of our global observation period most probably had entered the program before, and that delinquency on any previous loans therefore was not observed.

```
. craggit OdueOfDisbursedIfAbove10pc WasDelinq*pcPreviousLoan Overd* over* Year*
PovEntryPoor PovEntryUltra if include & IsLastLoan & mdy(9, 30, 2010) - MaturityDate >= 0
& mdy(9, 30, 2010) - LastCashTransactionDate > 90 & OpenDate > mdy(6, 30, 2005) ,
second(OdueOfDisbursedIfAbove10pc WasDelinq*pcPreviousLoan Overd* over* Year*
PovEntryPoor PovEntryUltra) vce(robust)

Estimating Cragg's tobit alternative
Assumes conditional independence

                                                Number of obs   =     143165
                                                Wald chi2(21)   =    9223.46
Log pseudolikelihood = -58658.875               Prob > chi2     =     0.0000

------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Tier1        |
WasDelinq1~n |   .2997822   .0138667    21.62   0.000      .272604    .3269604
WasDelinq3~n |   .3650791    .028973    12.60   0.000      .308293    .4218652
WasDelinq5~n |  -.0854674    .042371    -2.02   0.044    -.1685131   -.0024216
OverdGr1pcOD |   .2345619   .0119536    19.62   0.000     .2111332    .2579906
OverdGr2pcOD |   .1306276   .0147673     8.85   0.000     .1016842     .159571
OverdGr4pcOD |   .1283361   .0165206     7.77   0.000     .0959564    .1607158
OverdGr8pcOD |   .0011981    .022671     0.05   0.958    -.0432363    .0456325
OverdMFO1p~D |   .1784019   .0110056    16.21   0.000     .1568313    .1999724
OverdMFO2p~D |   .1999912   .0139794    14.31   0.000     .1725921    .2273903
OverdMFO4p~D |    .022497   .0218706     1.03   0.304    -.0203686    .0653627
OverdBr2pcOD |  -.0818889   .0095734    -8.55   0.000    -.1006524   -.0631254
OverdBr5pcOD |  -.1485128   .0137442   -10.81   0.000     -.175451   -.1215746
    over5000 |   .3452645   .0131864    26.18   0.000     .3194195    .3711094
   over10000 |   .0198121   .0094338     2.10   0.036     .0013221     .038302
   over20000 |  -.2452926   .0274215    -8.95   0.000    -.2990378   -.1915474
    Year2006 |  -.0006812   .0195572    -0.03   0.972    -.0390127    .0376503
    Year2007 |   -.269055   .0194336   -13.84   0.000    -.3071441   -.2309658
    Year2008 |   -.497771   .0195334   -25.48   0.000    -.5360558   -.4594863
Year2009or10 |  -.5234106    .019811   -26.42   0.000    -.5622395   -.4845817
PovEntryPoor |   .0024379   .0144572     0.17   0.866    -.0258976    .0307734
PovEntryUl~a |   .0519441   .0161393     3.22   0.001     .0203116    .0835765
       _cons |  -1.133414   .0249086   -45.50   0.000    -1.182234   -1.084594
-------------+----------------------------------------------------------------
Tier2        |
WasDelinq1~n |   .0514459   .0045135    11.40   0.000     .0425995    .0602922
WasDelinq3~n |   .0629706   .0077304     8.15   0.000     .0478193    .0781219
WasDelinq5~n |   .0385411   .0108953     3.54   0.000     .0171866    .0598956
OverdGr1pcOD |   .0144547   .0042056     3.44   0.001     .0062119    .0226975
OverdGr2pcOD |   .0241132   .0049455     4.88   0.000     .0144202    .0338063
OverdGr4pcOD |   .0052704    .005063     1.04   0.298    -.0046528    .0151937
OverdGr8pcOD |  -.0010485   .0064942    -0.16   0.872     -.013777    .0116799
OverdMFO1p~D |   .0023907   .0040646     0.59   0.556    -.0055758    .0103572
OverdMFO2p~D |   .0013418   .0047516     0.28   0.778    -.0079712    .0106547
OverdMFO4p~D |  -.0191263   .0066182    -2.89   0.004    -.0320977   -.0061548
OverdBr2pcOD |   .0082659    .003383     2.44   0.015     .0016354    .0148965
OverdBr5pcOD |  -.0149457   .0051985    -2.87   0.004    -.0251346   -.0047568
    over5000 |   .0403423    .005474     7.37   0.000     .0296135    .0510711
   over10000 |   .0113022    .003219     3.51   0.000     .0049931    .0176113
   over20000 |  -.0043734   .0106311    -0.41   0.681    -.0252101    .0164633
    Year2006 |   .0065786   .0061601     1.07   0.286    -.0054949    .0186521
    Year2007 |   .0111282    .006235     1.78   0.074    -.0010922    .0233486
    Year2008 |   -.024325   .0064837    -3.75   0.000    -.0370328   -.0116171
Year2009or10 |   .0079635   .0067388     1.18   0.237    -.0052443    .0211713
PovEntryPoor |   .0128367    .005244     2.45   0.014     .0025587    .0231147
PovEntryUl~a |  -.0444608   .0059223    -7.51   0.000    -.0560683   -.0328533
       _cons |   .3697368   .0088237    41.90   0.000     .3524428    .3870309
-------------+----------------------------------------------------------------
sigma        |
       _cons |   .2150603   .0009608   223.84   0.000     .2131772    .2169433
------------------------------------------------------------------------------
```

Predictions

```
. * predict see Stata Journal 9-4 p. 589

. predict x1pov if e(sample), eq(Tier1)

. predict x2pov if e(sample), eq(Tier2)

. predict sigmapov if e(sample), eq(sigma)

. gen ProbDefaultOver10pcPov = normal(x1pov)

. label var ProbDefaultOver10pcPov "Probability of default over 10 pc principal (model
with 2 poverty vars)"

. ren x2pov PredFractDefaultPov

. label var PredFractDefaultPov "Predicted fraction principal defaulted on (model with 2
poverty vars)"

. drop x1pov


.   summ     Odue100930ofDisbursed   OdueofDisbursedAbove10pc   OdueOfDisbursedIfAbove10pc
PredFractDefaultPov ProbDefaultOver10pcPov if include & IsLastLoan   & mdy(9, 30, 2010) -
MaturityDate >= 0 & mdy(9, 30, 2010) -  LastCashTransactionDate > 90 & OpenDate > mdy(6,
30, 2005)

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
Observed:
Odue100930~d |    158520    .0832894    .1936875  -.9691667          1
OdueofDisb~c |    158520    .1791572    .3834851          0          1
OdueOfDisb~c |    158520    .0809181    .1943728          0          1

Predicted:
PredFractD~v |    143165    .4226203    .0529677   .2788775   .6541405
ProbDefaul~v |    143165    .1849329    .1033468    .029565   .7807603
-------------+--------------------------------------------------------


. summ   OdueOfDisbursedIfAbove10pc if include & IsLastLoan    & mdy(9, 30, 2010) -
MaturityDate >= 0 & mdy(9, 30, 2010) -  LastCashTransactionDate > 90 & OpenDate > mdy(6,
30, 2005) & OdueOfDisbursedIfAbove10pc > 0

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
OdueOfDisb~c |     28400    .4516595    .2084035         .1          1
```

Adjusted predictions of default by poverty at entry

```
log:
C:\...\Analyses\Analyses110626_1_DefaultCORR\110626_1312AB_DefaultProbWomenAdjusted.log
```

We calculated adjusted probabilities, by poverty level at program entry from a probit model of defaulting on more than 10 percent of the principal. All other covariates were set to their means.

```
. adjust   WasDelinq10pcPreviousLoan WasDelinq30pcPreviousLoan WasDelinq50pcPreviousLoan
Overd* over* Year*  if include & IsLastLoan   & mdy(9, 30, 2010) -   MaturityDate >= 0 &
mdy(9, 30, 2010) -   LastCashTransactionDate > 90 & OpenDate > mdy(6, 30, 2005),
by( PovEntryPoor PovEntryUltra) pr

------------------------------------------------------------------------------------------------------------------------
     Dependent variable: OdueofDisb~c     Command: probit
 Covariates set to mean: WasDelinq1~n = .10635979, WasDelinq3~n = .0286313, WasDelinq5~n = .01116893,
                         OverdGr1pcOD = .36677261, OverdGr2pcOD = .22959522, OverdGr4pcOD = .11286278,
                         OverdGr8pcOD = .03581183, OverdMFO1p~D = .35581322, OverdMFO2p~D = .1578179,
                         OverdMFO4p~D = .03648937, OverdBr2pcOD = .42091293, OverdBr5pcOD = .13212727,
                         over5000 = .84804945, over10000 = .27932805, over20000 = .02606782, Year2006
                          = .16533371,
                         Year2007 = .22641009, Year2008 = .31348444, Year2009or10 = .24821709
------------------------------------------------------------------------------------------------------------------------

         ----------------------------
         Implied   |
```

```
poverty   | Implied poverty
level at  | level at entry:
entry:    |    ultra-poor
poor      |       0        1
----------+----------------
        0 | .164839  .178051
        1 | .165445
----------------------------
     Key:  Probability
```

The same was estimated for the model using gender instead of poverty level. We give the result here because we used it in the report:

```
----------------------
Customer  |
is member |
of        |
women's   |
borrower  |
group     |         pr
----------+-----------
        0 |    .152428
        1 |    .170904
----------------------
     Key:  pr  =  Probability
```

Borrower group-level analyses

Calculated variables for borrower group-level analyses

Recorded in the second half of
C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1500AB_VarsForBorrowerSurvey.log

and

C:\...\Analyses\Analyses110614_3_GroupLevelResults\110614_1819AB_HeterogeneityOverTime.log

```
.                                                                                use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep2010_tagsCORR_work07.dta"

. * Group level vars:
```

Group members ever

```
. * Number of group members ever in history of given group;

. gen byte temp = 1 if indivtag

. bysort GroupGId: egen temp2 = sum(temp)

.
. summ temp2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       temp2 |    948032    2905.588    10787.65          1      43364

.  * 43364 is the number of loans by borrowers without recorded group ID.
.
. gen  NoMembersEver = temp2
. label var   NoMembersEver "Number of individuals who ever received loans in a given group"


. drop temp temp2

.
. summ  NoMembersEver if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
```

```
NoMembersE~r |     16026    22.57569    11.55432          1         79
```

Date first loan opened in the group

```
. * Gen first date loan opened for anyone in given group:
.
. sort  GroupGId
. by  GroupGId: egen FirstLoanOpenInGroup = min(OpenDate)
```

Date of last transaction by any group member

```
. * Last transaction date in group:
.
. by GroupGId: egen LastTransactInGroup = max( LastTransactDate)
(214 missing values generated)
```

Length of (observed) partnership by the group

```
. gen LengthPartnerGroup =  LastTransactInGroup -  FirstLoanOpenInGroup
.
. summ  LengthPartnerGroup if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
LengthPart~p |     15994    1328.129    617.1893          1       2300


. summ  LengthPartnerGroup if grouptag, detail

        Days from first loan opened to last observed
                      transaction in group
-------------------------------------------------------------
      Percentiles      Smallest
 1%           38              1
 5%          259              2
10%          419              2       Obs               15994
25%          847              2       Sum of Wgt.       15994

50%         1463                      Mean           1328.129
                        Largest       Std. Dev.      617.1893
75%         1866           2300
90%         2076           2300       Variance       380922.6
95%         2160           2300       Skewness      -.3507024
99%         2284           2300       Kurtosis       1.950591
```

Individual membership and group size on certain calendar dates

```
. * Create "Was member at certain point in time", 1 July 2005 2007 2009
.
. * Membership defined by being between date first loan issued and date latest cash trans
last loan
.
.   gen   byte   MemberOn20050701   =   (   DateFirstLoanOpened   <=   mdy(7,1,2005)   &
LastCashTrDateByBorrower >=  mdy(7,1,2005))

.
. tab  MemberOn20050701 if indivtag, missing

MemberOn200 |
      50701 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    360,964       89.09       89.09
          1 |     44,198       10.91      100.00
------------+-----------------------------------
      Total |    405,162      100.00
```

Similarly for 1 July 2007 and 2009. Group size:

```
. * Number members in a group at a certain point in time:

.
. gen byte temp = (indivtag & MemberOn20050701)

.
. tab temp
```

```
       temp |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    903,834       95.34       95.34
          1 |     44,198        4.66      100.00
------------+-----------------------------------
      Total |    948,032      100.00


.
. by GroupGId: egen GrMembersOn20050701 = sum(temp)

.
. drop temp

.
. summ  GrMembersOn20050701 if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
GrM~20050701 |     16026    2.042369    4.870288          0         44

.
. replace  GrMembersOn20050701 = . if  GrMembersOn20050701 == 0
(581920 real changes made, 581920 to missing)

.
. summ  GrMembersOn20050701 if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
GrM~20050701 |      4336    7.548662    6.790348          1         44
```

and again analogously for 2007 and 2009. Used to compute transition tables.

Loan variables calculated at certain ages of group existence

```
. * Question whether over time within given group members become more heterogenous

. * See whether coeff.var. loan size differs first year, third year, fifth year.
.
. gen byte LoanFirstYearOfGroup = (OpenDate <=  FirstLoanOpenInGroup + 364)

. gen byte LoanThirdYearOfGroup = (OpenDate >   FirstLoanOpenInGroup + 364 + 365 ) &
(OpenDate <=  FirstLoanOpenInGroup + 364 + 2*365)

. gen byte LoanFifthYearOfGroup = (OpenDate >   FirstLoanOpenInGroup + 364 + 3*365 ) &
(OpenDate <=  FirstLoanOpenInGroup + 364 + 4*365)

. summ  LoanFirstYearOfGroup LoanThirdYearOfGroup LoanFifthYearOfGroup

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
LoanFirstY~p |    948032    .3291724    .4699129          0          1
LoanThirdY~p |    948032     .176839    .3815326          0          1
LoanFifthY~p |    948032    .0928745    .2902566          0          1

.
.
. by  GroupGId: egen NoLoansFirstYearOfGroup = sum( LoanFirstYearOfGroup)

. sum  NoLoansFirstYearOfGroup if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
NoLoansFir~p |     16026    18.75402     12.5709          1        111

. by  GroupGId: egen NoLoansThirdYearOfGroup = sum( LoanThirdYearOfGroup)

. sum  NoLoansThirdYearOfGroup if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
NoLoansThi~p |     16026    9.456571     8.87383          0         53

. by  GroupGId: egen NoLoansFifthYearOfGroup = sum( LoanFifthYearOfGroup)
```

```
. summ  NoLoansFifthYearOfGroup if grouptag

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
NoLoansFif~p |     16026    5.292025    7.813438          0         61

.
. bysort  GroupGId LoanFirstYearOfGroup: egen SumLoansFirstYearOfGroup =
sum( DisbursedAmtCPIadj )

. bysort  GroupGId LoanFirstYearOfGroup: egen SDLoansFirstYearOfGroup =
sd( DisbursedAmtCPIadj )

. bysort  GroupGId LoanThirdYearOfGroup: egen SumLoansThirdYearOfGroup =
sum( DisbursedAmtCPIadj )

. bysort  GroupGId LoanThirdYearOfGroup: egen SDLoansThirdYearOfGroup =
sd( DisbursedAmtCPIadj )

. bysort  GroupGId LoanFifthYearOfGroup: egen SumLoansFifthYearOfGroup =
sum( DisbursedAmtCPIadj )

. bysort  GroupGId LoanFifthYearOfGroup: egen SDLoansFifthYearOfGroup =
sd( DisbursedAmtCPIadj )

.
. replace  SumLoansFirstYearOfGroup = . if  LoanFirstYearOfGroup==0

. replace  SDLoansFirstYearOfGroup = . if  LoanFirstYearOfGroup==0

. replace  SumLoansThirdYearOfGroup = . if  LoanThirdYearOfGroup == 0

. replace  SDLoansThirdYearOfGroup = . if  LoanThirdYearOfGroup == 0

. replace  SumLoansFifthYearOfGroup = . if  LoanFifthYearOfGroup == 0

. replace  SDLoansFifthYearOfGroup = . if  LoanFifthYearOfGroup == 0

.
. gen  CVLoansFirstYearOfGroup =  (SDLoansFirstYearOfGroup *  NoLoansFirstYearOfGroup) /
SumLoansFirstYearOfGroup
(636552 missing values generated)

. gen  CVLoansThirdYearOfGroup =  (SDLoansThirdYearOfGroup *  NoLoansThirdYearOfGroup) /
SumLoansThirdYearOfGroup
(780858 missing values generated)

. gen  CVLoansFifthYearOfGroup =  (SDLoansFifthYearOfGroup *  NoLoansFifthYearOfGroup) /
SumLoansFifthYearOfGroup
(860342 missing values generated)

. summ CVL*

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
CVLoansFir~p |    311480     .318845    .1413063          0   2.370236
CVLoansThi~p |    167174    .4354882    .1805532          0   2.078416
CVLoansFif~p |     87690    .5355335    .2400584          0   2.110558


. count if  CVLoansFirstYearOfGroup ~=. &  CVLoansThirdYearOfGroup ~=. &
CVLoansFifthYearOfGroup ~= . & grouptag
    0

.
. * Thus these quantities have to be copied to all group members so that the record with
the grouptag finds them.
.
. gen temp1 =  CVLoansFirstYearOfGroup

. gen temp3 =  CVLoansThirdYearOfGroup

. gen temp5 =  CVLoansFifthYearOfGroup

. by  GroupGId: egen  CVLoansFirstYearOfGroup_b = mean(temp1)

. by  GroupGId: egen  CVLoansThirdYearOfGroup_b = mean(temp3)

. by  GroupGId: egen  CVLoansFifthYearOfGroup_b = mean(temp5)

. count if grouptag
```

```
16026

.  count  if    CVLoansFirstYearOfGroup_b  ~=.  &    CVLoansThirdYearOfGroup_b  ~=.  &
CVLoansFifthYearOfGroup_b ~= . & grouptag
 6675

.
. * Restrict to groups who finished their fifth year on or before our observation period:
.  count  if    CVLoansFirstYearOfGroup_b  ~=.  &    CVLoansThirdYearOfGroup_b  ~=.  &
CVLoansFifthYearOfGroup_b ~= . & grouptag & (OpenDate + 5* 365 + 1 <= mdy(9,30,2010))
 1447
```

In addition:

```
The skew of the distribution of CPI-adjusted loan principals in all loans in a given
group that were disbursed in the group's third year of operation:

.  by  GroupGId:  egen  SkewLoansThirdYearOfGroup  =  skew(DisbursedAmtCPIadj)  if
LoanThirdYearOfGroup
```

Group lifetime

Observed

Definition of variable, see above.

```
. summ  LengthPartnerGroup if grouptag, detail

        Days from first loan opened to last observed
                    transaction in group
-------------------------------------------------------------
      Percentiles      Smallest
 1%           38              1
 5%          259              2
10%          419              2       Obs               15994
25%          847              2       Sum of Wgt.       15994

50%         1463                      Mean           1328.129
                        Largest       Std. Dev.      617.1893
75%         1866           2300
90%         2076           2300       Variance       380922.6
95%         2160           2300       Skewness      -.3507024
99%         2284           2300       Kurtosis       1.950591
```

Estimated from survival model

```
log:
C:\...\Analyses\Analyses110627_1_GroupLifespanEstimate\110627_0914AB_GroupLifespanEstimat
e.log

. gen byte GroupLifeFailureEvent = ( LastTransactInGroup < mdy(7,1,2010))

.  label var  GroupLifeFailureEvent "Last cash transaction in group was before 1st July
2010"

. tab  GroupLifeFailureEvent if grouptag & include

  Last cash |
transaction |
   in group |
 was before |
   1st July |
       2010 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |     12,955       81.60       81.60
          1 |      2,922       18.40      100.00
------------+-----------------------------------
      Total |     15,877      100.00


. stset   LastTransactInGroup if grouptag & include  ,  failure( GroupLifeFailureEvent==1)
time0( FirstLoanOpenInGroup) exit(time .) origin(time  FirstLoanOpenInGroup)

------------------------------------------------------------------------------
   948032  total obs.
```

```
   932155  ignored at outset because of -if <exp>-
       32  event time missing (LastTransactInGroup>=.)          PROBABLE ERROR
------------------------------------------------------------------------------
    15845  obs. remaining, representing
     2922  failures in single record/single failure data
 2.11e+07  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      2300

. stdes

                                |-------------- per subject --------------|
Category                  total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects           15845
no. of records            15845           1           1          1          1

(first) entry time                        0           0          0          0
(final) exit time                  1332.443           1       1470       2300

subjects with gap             0
time on gap if gap            0
time at risk           21112566    1332.443           1       1470       2300

failures                   2922    .1844115           0          0          1
------------------------------------------------------------------------------

. stsum

         |               incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+--------------------------------------------------------------------
   total |     21112566   .0001384         15845       1855         .         .

Exponential extrapolation:

. stci, emean

         failure _d:  GroupLifeFailureEvent == 1
   analysis time _t:  (LastTransactInGroup-origin)
             origin:  time FirstLoanOpenInGroup
  exit on or before:  time .

             |    no. of    extended
             |  subjects        mean
-------------+----------------------
       total |     15845    6106.151

. di 6106/364.25
16.763212 years
```

Changes in within-group loan size diversity

```
. * Summaries of coefficients of variation of size of loans issued during first, third
and fifth year of group existence:
```

For all groups

```
.
. summ  CVLoansFirstYearOfGroup_b if grouptag, detail

                  CVLoansFirstYearOfGroup_b
-------------------------------------------------------------
      Percentiles      Smallest
 1%     .0034019              0
 5%     .1076526              0
10%     .1595581              0       Obs               15440
25%     .2243704              0       Sum of Wgt.       15440

50%     .2983351                      Mean           .3224151
                        Largest       Std. Dev.      .1607473
75%      .395138       1.573267
90%     .5185227       1.801509       Variance       .0258397
95%     .6108751        1.89284       Skewness       1.481685
99%     .8431835       2.370236       Kurtosis       9.524958

. summ  CVLoansThirdYearOfGroup_b if grouptag, detail
```

```
                  CVLoansThirdYearOfGroup_b
-------------------------------------------------------------
      Percentiles      Smallest
 1%     .0415168              0
 5%     .1720002              0
10%     .2250234              0       Obs               11229
25%     .2990991              0       Sum of Wgt.       11229

50%     .3871149                      Mean           .4190514
                        Largest       Std. Dev.      .1939739
75%     .4992986       1.606345
90%     .6565977       1.765356       Variance       .0376259
95%      .781486       1.831347       Skewness       1.451746
99%     1.077404       2.078416       Kurtosis       7.509346

. summ  CVLoansFifthYearOfGroup_b if grouptag, detail

                  CVLoansFifthYearOfGroup_b
-------------------------------------------------------------
      Percentiles      Smallest
 1%     .0385927              0
 5%     .1850028              0
10%     .2548136              0       Obs                6875
25%     .3505361              0       Sum of Wgt.        6875

50%     .4615377                      Mean           .5129059
                        Largest       Std. Dev.      .2567168
75%     .6211138        1.95584
90%     .8475955       1.964537       Variance       .0659035
95%     1.011138       2.108002       Skewness       1.320448
99%     1.372043       2.110558       Kurtosis       6.034743
```

For groups active for five years before 30 September 2010

```
. summ  CVLoansFirstYearOfGroup_b CVLoansThirdYearOfGroup_b CVLoansFifthYearOfGroup_b if
CVLoansFirstYearOfGroup_b>         ~=.    &         CVLoansThirdYearOfGroup_b    ~=.    &
CVLoansFifthYearOfGroup_b ~= . & grouptag & (OpenDate + 4* 365 + 1 <= mdy(9,30,2010)),
detail

                  CVLoansFirstYearOfGroup_b
-------------------------------------------------------------
      Percentiles      Smallest
 1%      .113897              0
 5%      .178606              0
10%     .2085695         .00357       Obs                3248
25%     .2556449        .0122128      Sum of Wgt.        3248

50%     .3298475                      Mean           .3666585
                        Largest       Std. Dev.      .1598362
75%     .4447798        1.20537
90%     .5726382       1.253293       Variance       .0255476
95%     .6649536       1.350129       Skewness       1.426627
99%     .8878802       1.473237       Kurtosis       6.544366

                  CVLoansThirdYearOfGroup_b
-------------------------------------------------------------
      Percentiles      Smallest
 1%     .1285029              0
 5%     .1981222              0
10%     .2335746       .0091068       Obs                3248
25%     .3002023       .0172525       Sum of Wgt.        3248

50%     .3869435                      Mean           .4231478
                        Largest       Std. Dev.      .1908538
75%     .5012393       1.527375
90%     .6651883       1.588326       Variance       .0364252
95%     .7834553       1.765356       Skewness       1.717581
99%     1.040207       2.078416       Kurtosis       8.958341

                  CVLoansFifthYearOfGroup_b
-------------------------------------------------------------
      Percentiles      Smallest
 1%     .0705848              0
 5%     .1924501              0
10%     .2614826              0       Obs                3248
25%     .3521567              0       Sum of Wgt.        3248
```

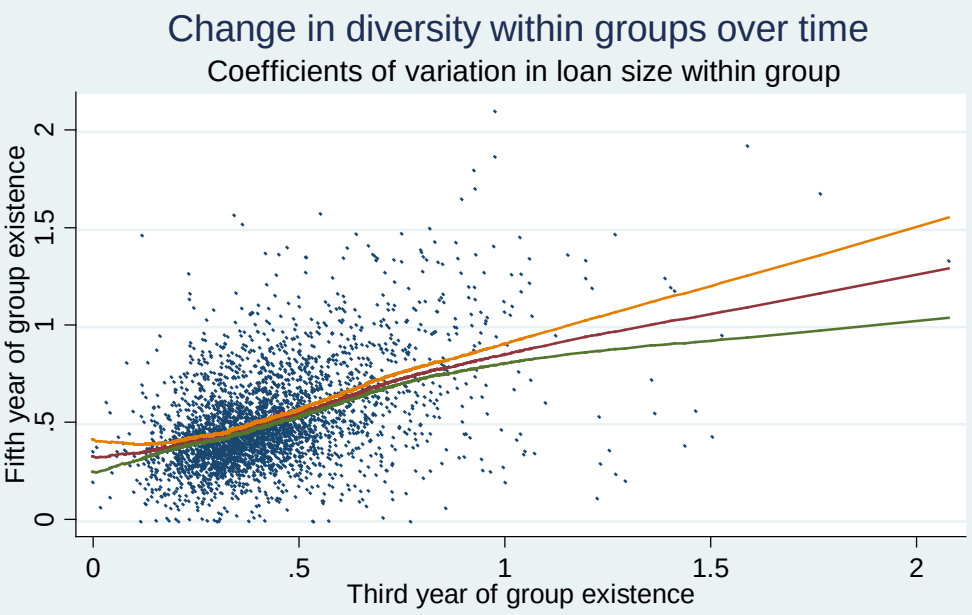
```
50%     .4587919                    Mean           .5080172
                      Largest       Std. Dev.      .2475688
75%     .6105488      1.803913
90%     .8321478      1.874342      Variance       .0612903
95%     .9776399      1.928891      Skewness       1.347961
99%     1.356531      2.108002      Kurtosis       6.194014
```

Regressions of loan size coeff.var. on values of two years before

```
.
. regress  CVLoansThirdYearOfGroup_b CVLoansFirstYearOfGroup_b  if
CVLoansFirstYearOfGroup_b ~=. &  CVLoansThirdYearOfGroup_b ~=. &
CVLoansFifthYearOfGroup_b ~= . & grouptag & (OpenDate + 4* 365 + 1 <= mdy(9,30,2010))

      Source |       SS       df       MS              Number of obs =    3248
-------------+------------------------------           F(  1,  3246) =  904.21
       Model |  25.7680965     1  25.7680965           Prob > F      =  0.0000
    Residual |  92.5043859  3246  .028497962           R-squared     =  0.2179
-------------+------------------------------           Adj R-squared =  0.2176
       Total |  118.272482  3247  .036425156           Root MSE      =  .16881

------------------------------------------------------------------------------
CVLoansThi~b |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
CVLoansFir~b |    .557346   .0185349    30.07   0.000     .5210046    .5936873
       _cons |   .2187922   .0074135    29.51   0.000     .2042566    .2333277
------------------------------------------------------------------------------


. regress  CVLoansFifthYearOfGroup_b  CVLoansThirdYearOfGroup_b   if
CVLoansFirstYearOfGroup_b ~=. &  CVLoansThirdYearOfGroup_b ~=. &
CVLoansFifthYearOfGroup_b ~= . & grouptag & (OpenDate + 4* 365 + 1 <= mdy(9,30,2010))

      Source |       SS       df       MS              Number of obs =    3248
-------------+------------------------------           F(  1,  3246) =  927.96
       Model |  44.2441462     1  44.2441462           Prob > F      =  0.0000
    Residual |  154.765437  3246  .047678816           R-squared     =  0.2223
-------------+------------------------------           Adj R-squared =  0.2221
       Total |  199.009583  3247  .061290294           Root MSE      =  .21835

------------------------------------------------------------------------------
CVLoansFif~b |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
CVLoansThi~b |   .6116262    .020078    30.46   0.000     .5722593     .650993
       _cons |   .2492089   .0093199    26.74   0.000     .2309354    .2674825
------------------------------------------------------------------------------
```

For the total population of groups in business for at least five years, the diversity in active loan sizes increased. However, the coefficient on the previous value is positive, but smaller than one, in both period comparisons. This means that for the individual groups there is a regression to the mean.

Figure 5: Change in diversity within groups over time



Note: 3,248 groups with loans in their first, third and fifth years, and who completed their fifth year. All loans opened during year in point. CPI-adjusted. Local regression, 95%CI. R-sq = 0.23. Mean Coeff.Var. - First year = 0.37, third = 0.42, fifth = 0.51.

Loans in the fifth year of operation in response to third-year characteristics

A zero-inflated negative binomial model was estimated of the number of all loan disbursed to group members in the fifth year of operation. The count was regressed on third-year loan distribution characteristics.

```
      log:
C:\Users\Aldo\Documents\Aldo\RDRS\RDRS2011a\Analyses\Analyses110822_1_LoanPrincipCVandPau
l\110822_2016AB_LoansFifthYearOfGroup_ZinbModel.log

.                                                                                save
"C:\Users\Aldo\Documents\Aldo\RDRS\RDRS2011a\Analyses\Analyses110822_1_LoanPrincipCVandPa
ul\110822_1944AB_LoanFileReducedToNoLoans5thYear_ZinbModel.dta", replace

. . * A group tag must be recreated such that each group with calculated skew for the
third-year loan principals was included once. This was done before reducing the loan file
to the above. The command was:

. * egen grouptagalt2 = tag(GroupGId) if  SkewLoansThirdYearOfGroup ~=.


. count
150654

. * Number of groups with 3rd year activity:

. count if  grouptagalt2
11143

. * Variables:

.  des    NoLoansFifthYearOfGroup    NoLoansThirdYearOfGroup    MeanLoansThirdYearOfGroup
CVLoansThirdYearOfGroup SkewLoansThirdYearOfGroup grouptagalt2 LengthPartnerGroup
```

```
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------
NoLoansFifthY~p float  %9.0g                 Number loans taken by group in its fifth year
NoLoansThirdY~p float  %9.0g                 Number loans taken by group in its third year
MeanLoansThir~p float  %9.0g                 Mean of loans issued in third year of group (CPI-adj.)
CVLoansThirdY~p float  %9.0g                 Coeff. var. loan size within group in its third year
SkewLoansThir~p float  %9.0g                 Skew of loan size within groups active in third year
grouptagalt2    byte   %8.0g                  Tag for groups with non-blank skew of intra-group third-year
loan size
LengthPartner~p float  %9.0g                   Days from first loan opened to last observed transaction in
group

. summ    NoLoansFifthYearOfGroup    NoLoansThirdYearOfGroup    MeanLoansThirdYearOfGroup
CVLoansThirdYearOfGroup       SkewLoansThirdYearOfGroup       if       grouptagalt2      &
LengthPartnerGroup >= 3

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
NoLoansFif~p |     11143    7.530468    8.386954          0         61
NoLoansThi~p |     11143    13.52006    7.655193          2         53
MeanLoansT~p |     11143    10264.62    3951.515   2797.623   84477.74
CVLoansThi~p |     11143    .4210494    .1920543   .0018602   2.078416
SkewLoansT~p |     11143    .5805626    .8634556  -4.407688   5.356765

. * Dead group fifth year:

. count if  grouptagalt2 & NoLoansFifthYearOfGroup == 0
 4054

. di 4054 / 11143
.36381585


. * Zero inflated negative binomial model:

.  zinb    NoLoansFifthYearOfGroup    NoLoansThirdYearOfGroup    MeanLoansThirdYearOfGroup
CVLoansThirdYearOfGroup     SkewLoans     ThirdYearOfGroup     if     grouptagalt2     &
LengthPartnerGroup  >=  3,  inflate(NoLoansThirdYearOfGroup    MeanLoansThirdYearOfGroup
CVLoansThirdYearOfGroup SkewLoansThirdYearOfGroup)


Zero-inflated negative binomial regression        Number of obs   =      11143
                                                  Nonzero obs     =       7089
                                                  Zero obs        =       4054

Inflation model = logit                           LR chi2(4)      =    1334.60
Log likelihood  = -29632.81                       Prob > chi2     =     0.0000

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
NoLoansFif~p |
NoLoansThi~p |   .0379544   .0010432    36.38   0.000     .0359098     .039999
MeanLoansT~p |  -8.06e-06   2.13e-06    -3.78   0.000    -.0000122   -3.88e-06
CVLoansThi~p |   .1161114   .0575702     2.02   0.044     .0032758    .228947
SkewLoansT~p |   -.025912   .0116482    -2.22   0.026    -.0487422   -.0030819
       _cons |   1.891884   .0303814    62.27   0.000     1.832337    1.95143
-------------+----------------------------------------------------------------
inflate      |
NoLoansThi~p |   -.098698   .0034386   -28.70   0.000    -.1054375   -.0919585
MeanLoansT~p |  -.0000618   6.78e-06    -9.11   0.000    -.0000751   -.0000485
CVLoansThi~p |  -.7296451   .1571348    -4.64   0.000    -1.037624   -.4216666
SkewLoansT~p |  -.1204261   .0334739    -3.60   0.000    -.1860337   -.0548184
       _cons |   1.649794   .0884877    18.64   0.000     1.476362    1.823227
-------------+----------------------------------------------------------------
    /lnalpha |  -1.173352   .0245139   -47.86   0.000    -1.221399   -1.125306
-------------+----------------------------------------------------------------
       alpha |   .3093282   .0075828                      .2948175    .3245531
------------------------------------------------------------------------------

. predict predLoansInGroupYear5 if e(sample)
(option n assumed; predicted number of events)

(139511 missing values generated)

. label var  predLoansInGroupYear5 "Predicted number of loans issued in fifth year"


.  spearman     NoLoansFifthYearOfGroup       predLoansInGroupYear5   if   grouptagalt2   &
LengthPartnerGroup >= 3
```

```
 Number of obs =   11143
Spearman's rho =       0.4529

Test of Ho: NoLoansFifthYearOf~p and predLoansInGroupYe~5 are independent
    Prob > |t| =       0.0000


.  spearman     NoLoansFifthYearOfGroup       predLoansInGroupYear5  if  grouptagalt2  &
LengthPartnerGroup >= 3 & NoLoansFift
> hYearOfGroup    > 0

 Number of obs =    7089
Spearman's rho =       0.4117

Test of Ho: NoLoansFifthYearOf~p and predLoansInGroupYe~5 are independent
    Prob > |t| =       0.0000


. mfx, eyex

Elasticities after zinb
      y  = predicted number of events (predict)
         =  7.0412688
------------------------------------------------------------------------------
variable |      ey/ex    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
N~Thir~p |   .9569496      .02045   46.80   0.000   .916869   .99703   13.5201
M~Thir~p |   .1283748       .0307    4.18   0.000   .068207  .188543   10264.6
C~Thir~p |   .1510646      .03207    4.71   0.000   .088206  .213924   .421049
SkewLo~p |   .0082091      .00921    0.89   0.373  -.009846  .026265   .580563
------------------------------------------------------------------------------
```

Borrower survey

Original survey data

Pre-processing

Source:
C:\...\Analyses\Analyses110616_2_BorrowerSurvey\110616_1649AB_BorrowerSurvey_work09.dta"

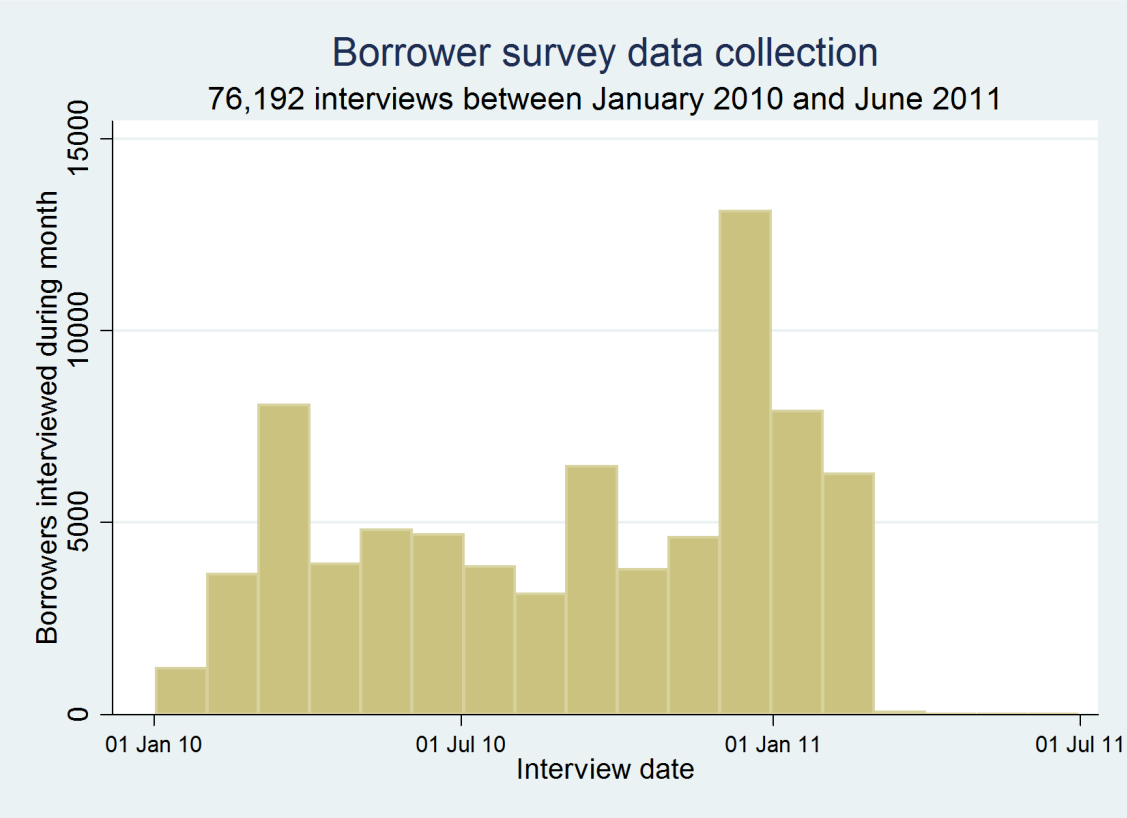
Clean-up and recodings logged in:

C:\...\Analyses\Analyses110616_2_BorrowerSurvey\110616_1648AB_BorrowerSurveyReformat.log

and

C:\...\Analyses\Analyses110616_2_BorrowerSurvey\110617_1408AB_DurableConsumer_MissingToZe
ro.log

Figure 6: Time of borrower interviews, 2010-11



Household wealth scale

(Mokken 1971; Benini 2007; Hardouin 2007)

```
log:  C:\...\Analyses\Analyses110617_1_MokkenScale\110617_1435AB_FirstMokkenAndRasch.log
```

Mokken scale

```
. mokken Own*
IRT  Non-parametric Mokken scaling

Variable               |  Obs    %Pos      Hi     z(H)    Label
---------+-----------------------------------------------------
OwnMotorBike           |76192   0.027   0.640  120.230
OwnSewingMachine       |76192   0.045   0.465  104.393
OwnTV                  |76192   0.176   0.681  252.745
OwnElectricity         |76192   0.258   0.542  231.332
OwnAlmirah             |76192   0.308   0.530  233.017
OwnTelephone           |76192   0.315   0.496  216.954
OwnWatchOrClock        |76192   0.608   0.617  199.713
OwnChairOrBench        |76192   0.802   0.741  178.012
OwnCotOrBed            |76192   0.977   0.647   62.534
---------+-----------------------------------------------------
Mokken H               |                0.579  395.165

According to Mokken (1971:185), H>=0.50 is a strong scale
```

Rasch scale

```
. raschtest Own*, id( recno) nodraw genlt(OwnLt) genscore(OwnScore)

Estimation method: Conditional maximum likelihood (CML)
Number of items: 9
Number of groups: 10 (8 of them are used to compute the statistics of test)
```

```
Number of individuals: 76192
Number of individuals with missing values: 0 (removed)
Number of individuals with nul or perfect score: 1477
Conditional log-likelihood: -96209.3514
Log-likelihood: -155821.6155
```

Items	Difficulty parameters	std Err.	R1c	df	p-value	Standardized Outfit	Infit	U
OwnCotOrBed	-1.1e+01	0.049711	101.111	7	0.0000	9.911	0.848	24.773
OwnChairOr~h	-7.64549	0.03262	281.295	7	0.0000	5.550	-8.675	4.213
OwnAlmirah	-4.02304	0.02970	313.473	7	0.0000	-4.015	-5.478	-4.635
OwnSewingM~e	-0.63595	0.031723	011.604	7	0.0000	7.471	14.008	13.766
OwnWatchOr~k	-6.00696	0.030379	711.809	7	0.0000	6.138	1.555	12.991
OwnElectri~y	-3.61986	0.02968	938.295	7	0.0000	0.215	-5.040	0.581
OwnTV	-2.80390	0.029642	231.786	7	0.0000	-7.948	-38.393	-22.796
OwnTelephone	-4.07777	0.029711	423.696	7	0.0000	9.278	13.808	15.840
OwnMotorBi~*	0.00000	.	1021.620	7	0.0000	2.990	-4.963	2.156
R1c test			R1c=6835.530	56	0.0000			
Andersen LR test			Z=5393.693	56	0.0000			

*: The difficulty parameter of this item had been fixed to 0

Group	Score	Ability parameters	std Err.	Freq.	Expected Score	ll
0	0	-10.000	1.790	1085	0.82	
1	1	-9.157	1.730	9905	1.09	-1448.5477
2	2	-6.906	1.292	13332	2.11	-11491.3870
3	3	-5.476	1.097	18964	3.12	-21769.6293
4	4	-4.454	0.979	12548	4.08	-24304.8011
5	5	-3.625	0.951	7851	4.98	-17684.4860
6	6	-2.763	1.004	5124	5.90	-11084.4798
7	7	-1.608	1.131	5279	6.92	-4200.4496
8	8	-0.245	1.258	1712	7.88	-1528.7240
9	9	1.383	1.842	392	8.65	

Import from loan table

Variables

Additional variables calculated

Using

```
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"

. * Ultimately, for the merging into the borrower survey table, one record per borrower
from this table only will be needed.
. * This reduction has not yet taken place here.

. * Vars on first loan, last loan, group status, group loan characteristics

. * ever delinquent, and at which level [not yet done here].

. * Last loan:

. gsort  IndividualCustomerGId -LoanSeq

. egen IsLastLoan = tag( IndividualCustomerGId)
. label var  IsLastLoan "Last loan in customer history"

. tab  IsLastLoan IsFirstLoan
```

```
Last loan  |     First loan in
in custome |   customer history
history    |         0          1 |     Total
-----------+----------------------+----------
         0 |   297,722    245,148 |   542,870
         1 |   245,148    160,014 |   405,162
-----------+----------------------+----------
     Total |   542,870    405,162 |   948,032


. clonevar OpenDateLastLoan = OpenDate

.  replace     OpenDateLastLoan   =     OpenDateLastLoan[_n-1]   if     IsLastLoan==0   &
IndividualCustomerGId == IndividualCustomerGId[_n-1]

. label var  OpenDateLastLoan "Date last loan disbursed in customer history"

. clonevar  DisbursedAmtCPIadjLastLoan = DisbursedAmtCPIadj

.  replace       DisbursedAmtCPIadjLastLoan  =       DisbursedAmtCPIadjLastLoan[_n-1]  if
IsLastLoan==0 &  IndividualCustomerGId == IndividualCustomerGId[_n-1]

.  label  var   DisbursedAmtCPIadjLastLoan  "Principal  last  loan  (CPI  adj.)  in  customer
history"

. clonevar  logPrincipCPILastLoan =  logPrincipCPI

. replace     logPrincipCPILastLoan=     logPrincipCPILastLoan[_n-1] if   IsLastLoan==0 &
IndividualCustomerGId == IndividualCustomerGId[_n-1]

.  label var   logPrincipCPILastLoan "Principal last loan (log10 - CPI adj.) in customer
history"


. * Total sum of loans:

. by  IndividualCustomerGId: egen SumAllLoansCPIadj = sum(  DisbursedAmtCPIadj)

. label var  SumAllLoansCPIadj "Sum all loan amounts disbursed - CPI adj."


.  * Construct dummy if last loan repaid:
.
. gen byte temp = (IsLastLoan &  LoanRepaidDate ~= .)

. by  IndividualCustomerGId: egen LastLoanIsRepaid = max(temp)

. drop temp

. label var  LastLoanIsRepaid "Customer has repaid last loan"
```

Import

Source:

```
use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"
```

Reduced to smaller file with 87 variables:

```
save
"C:\...\Analyses\Analyses110616_1_LoanTableEdits\110616_1300AB_DataToMergeToBorrowerSurve
y_work08.dta"
```

Later these variables were also added, through a separate merge operation:

```
-------------------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------------------
OpenDateFirst~n int     %tdD_m_Y                  Date first loan disbursed in customer
history
DisbursedAmtC~n float  %9.0g                 Principal first loan (CPI adj.) in customer
history
logPrincipCPI~n float  %9.0g                  Principal first loan (log10 - CPI adj.) in
customer history
-------------------------------------------------------------------------------------------
```

Merge operation

```
. merge  IndividualCustomerGId using
"C:\...\Analyses\Analyses110616_1_LoanTableEdits\110616_1300AB_DataToMergeToBorrowerSurve
y_work08.dta"

. tab _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |      9,400        2.27        2.27
          2 |    338,370       81.62       83.89
          3 |     66,792       16.11      100.00
------------+-----------------------------------
      Total |    414,562      100.00

. drop if _merge == 2

. gen byte HasLoanData = (_merge == 3)

. drop _merge
```

Saved file:

```
save
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1603AB_BorrowerSurvey_and_
LoanInfo_work10.dta"
```

Analytic weights

Marking loans in loan table that have borrowers with survey data

Goes back to loan table

Auxiliary file

```
.                                                                                   use
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1603AB_BorrowerSurvey_and_
LoanInfo_work10.dta", clear

. keep if  HasLoanData

. sort  IndividualCustomerGId

.                                                                                 save
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1935AB_AuxilListAllClients
WithBothLoanAndSurveyData.dta", replace
```

Merging back

```
.                                                                                   use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCO
> RR_work07.dta"

. sort  IndividualCustomerGId

.              merge                                IndividualCustomerGId              using
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1935AB_AuxilListAllClients
WithBothLoanAndSurveyData.dta"

. tab  HasLoanData, missing

HasLoanData |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |    196,115       20.69       20.69
          . |    751,917       79.31      100.00
------------+-----------------------------------
      Total |    948,032      100.00

. ren  HasLoanData InSurveySample

. replace  InSurveySample = 0 if  InSurveySample == .

. drop  _merge
```

```
. tab  InSurveySample if indivtag

InSurveySam |
        ple |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    338,370       83.51       83.51
          1 |     66,792       16.49      100.00
------------+-----------------------------------
      Total |    405,162      100.00
```

Logit model of inclusion in borrower survey sample

In the model here, i.U_id is the binarized set of RDRS Program Units, most of them co-extensive with a district. The other variables are self-explanatory.

```
.  xi:  logit    InSurveySample  i.U_id    GroupIsFemale    PovEntryPoor  PovEntryUltra
MemberOn20050701      MemberOn20070701      MemberOn20090701      CustomerEverDelinq30pc
logPrincipCPILastLoan if indivtag & IsFirstLoan
i.U_id            _IU_id_0-9          (naturally coded; _IU_id_0 omitted)

Logistic regression                               Number of obs   =     358671
                                                  LR chi2(17)     =   18722.56
                                                  Prob > chi2     =     0.0000
Log likelihood = -161327.45                       Pseudo R2       =     0.0548

------------------------------------------------------------------------------
InSurveySa~e |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    _IU_id_1 |  -.4411372   .0292679   -15.07   0.000    -.4985013   -.3837732
    _IU_id_2 |  -.3838096   .0279252   -13.74   0.000     -.438542   -.3290773
    _IU_id_3 |  -.3511803   .0305642   -11.49   0.000    -.4110851   -.2912756
    _IU_id_4 |  -.6662122   .0272915   -24.41   0.000    -.7197024   -.6127219
    _IU_id_5 |  -.8149764   .0274542   -29.68   0.000    -.8687856   -.7611672
    _IU_id_6 |  -.9366606   .0277985   -33.69   0.000    -.9911446   -.8821765
    _IU_id_7 |  -.9102714   .0286796   -31.74   0.000    -.9664824   -.8540605
    _IU_id_8 |  -.8694914   .0277729   -31.31   0.000    -.9239253   -.8150576
    _IU_id_9 |  -.1486427   .0280322    -5.30   0.000    -.2035848   -.0937007
GroupIsFem~e |   .3397714   .0167062    20.34   0.000     .3070278    .3725149
PovEntryPoor |   .1642408   .0180097     9.12   0.000     .1289425    .1995391
PovEntryUl~a |   .3695094   .0199261    18.54   0.000     .3304549    .4085638
Mem~20050701 |   -.220107   .0183744   -11.98   0.000    -.2561202   -.1840938
Mem~20070701 |  -.4340156   .0107278   -40.46   0.000    -.4550416   -.4129895
Mem~20090701 |   .7410306   .0099767    74.28   0.000     .7214766    .7605846
Custome~30pc |  -.6836739   .0160195   -42.68   0.000    -.7150716   -.6522762
logPrincip~n |   .9054506   .0222206    40.75   0.000     .8618991    .9490021
       _cons |   -5.13292   .0946895   -54.21   0.000    -5.318508   -4.947332
------------------------------------------------------------------------------

. predict  probInSampleFull
(option pr assumed; Pr(InSurveySample))

. summ  probInSampleFull

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
probInSamp~l |    880908    .1960772    .0881321   .0046258   .6213306
```

Creating analytic weights in the borrower survey table

Auxiliary file to merge back to survey table

```
.                                                                                use
"C:\...\Analyses\Analyses110614_2_RecalculateTags\110614_1302AB_IndivLoans_Jun2004_Sep201
0_tagsCORR_work07.dta"

.                                                                                save
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110618_0642AB_AuxilInSampleProbLo
gitLoanTable.dta", replace
. keep if indivtag

. keep  IndividualCustomerGId  probInSampleFull

. sort  IndividualCustomerGId
```

```
. save
```

Merge operation

```
. use
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1603AB_BorrowerSurvey_and_
LoanInfo_work10.dta"

. sort  IndividualCustomerGId

.              merge                              IndividualCustomerGId              using
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110618_0642AB_AuxilInSampleProbLo
gitLoanTable.dta"

. tab _merge

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |      9,400        2.27        2.27
          2 |    338,370       81.62       83.89
          3 |     66,792       16.11      100.00
------------+-----------------------------------
      Total |    414,562      100.00

. drop if _merge == 2

. tab  _merge HasLoanData

           |      HasLoanData
    _merge |         0          1 |     Total
-----------+----------------------+----------
         1 |     9,400          0 |     9,400
         3 |         0     66,792 |    66,792
-----------+----------------------+----------
     Total |     9,400     66,792 |    76,192


. drop _merge
```

Analytic weights

```
. gen  pwFullModel = 1 / probInSampleFull

. summ  pwFullModel pwModelNoYears

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
 pwFullModel |     65632     5.31987    2.611826   1.613406   74.55878
```

Survey estimation, with *svyset* was not done. Instead, *pwFullModel* was treated as an analytic weight. At this point, we are only interested in differences in the estimates of the key outcome variable, the number of durable consumer goods owned by the surveyed borrowers, calculated (see above) as *OwnRaschScore*.

```
. tab  OwnRaschScore

OwnRaschSco |
         re |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,085        1.42        1.42
          1 |      9,905       13.00       14.42
          2 |     13,332       17.50       31.92
          3 |     18,964       24.89       56.81
          4 |     12,548       16.47       73.28
          5 |      7,851       10.30       83.58
          6 |      5,124        6.73       90.31
          7 |      5,279        6.93       97.24
          8 |      1,712        2.25       99.49
          9 |        392        0.51      100.00
------------+-----------------------------------
      Total |     76,192      100.00

. tab  OwnRaschScore [aw =  pwFullModel]

OwnRaschSco |
```

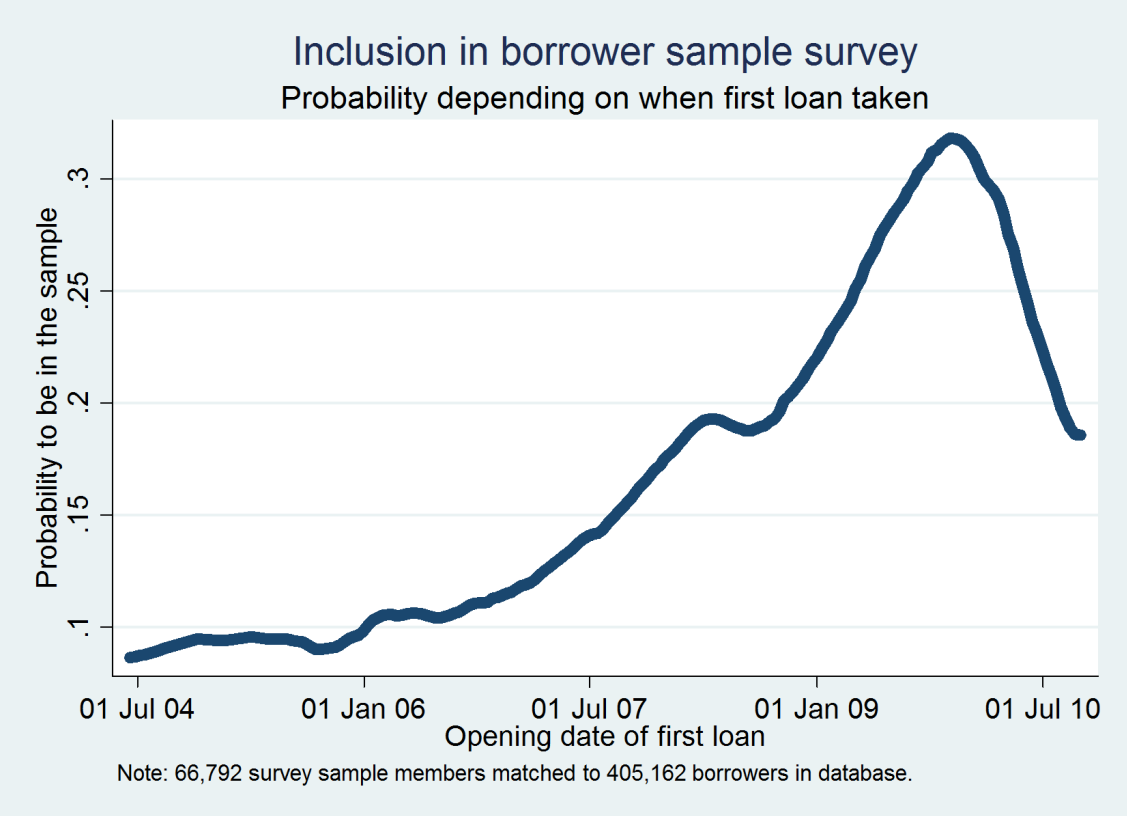
```
         re |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |  921.674421        1.40        1.40
          1 |  9,097.9608       13.86       15.27
          2 |  11,746.884       17.90       33.16
          3 |  16,518.233       25.17       58.33
          4 |  10,534.544       16.05       74.38
          5 |  6,681.2141       10.18       84.56
          6 |  4,151.9542        6.33       90.89
          7 |  4,271.9754        6.51       97.40
          8 |  1,376.7745        2.10       99.50
          9 |  330.785376        0.50      100.00
------------+-----------------------------------
      Total |     65,632      100.00
```

```
. summ  OwnRaschScore

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
OwnRaschSc~e |     76192    3.515185    1.896159          0          9
```

```
. summ  OwnRaschScore [aw =  pwFullModel]

    Variable |     Obs      Weight        Mean   Std. Dev.       Min        Max
-------------+-----------------------------------------------------------------
OwnRaschSc~e |   65632  349153.724    3.451022   1.881713          0          9
```

The differences are minimal.

Figure 7: Probability for a borrower to be included in the survey



Note: Graph uses smooths of predicted probability from a local regression model. Not detailed here.

Household wealth among surveyed borrowers

We estimated a model of household wealth, measured as the proportion of 9 durable consumer goods owned (see above), in response to poverty and borrower career factors. The sample comprised of a subset of borrower survey households that were group members with active loans on 1st July 2005. The limitations of this approach - selection mechanisms and drop-outs were not observed - are discussed in the study report.

Estimation was via a logit model in GLM (Baum 2008).

Source:

```
log:
C:\...\Analyses\Analyses110622_3_BorrowerSurvey\110622_1631AB_RaschScore_GLM_proport.log

.                                                                                                use
"C:\...\Analyses\Analyses110617_3_MergeLoanVarsToSurvey\110617_1603AB_BorrowerSurvey_and_
LoanInfo_work10.dta"
```

Variables

```
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------------------------------------
RaschScore      float  %9.0g                 Score on Rasch scale of 9 household items
SDLoansFirstY~p float  %9.0g                 SD of loans issues in first year of group (CPI-adj.)
SDLoansFifthY~p float  %9.0g                 SD of loans issues in fifth year of group (CPI-adj.)
logPrin~rstLoan float  %9.0g                 Principal first loan (log10 - CPI adj.) in customer history
logPrin~astLoan float  %9.0g                 Principal last loan (log10 - CPI adj.) in customer history
PovEntryPoor    byte   %8.0g                 Implied poverty level at entry: poor
PovEntryUltra   byte   %8.0g                 Implied poverty level at entry: ultra-poor
CustomerEv~10pc float  %9.0g                 Customer was at some point delinquent with 10 pc of principal
CustomerEv~30pc float  %9.0g                 Customer was at some point delinquent with 30 pc of principal
CustomerEv~50pc float  %9.0g                 Customer was at some point delinquent with 50 pc of principal
pwFullModel     float  %9.0g                 Reciprocal of probability of being in survey sample
Member~20050701 byte   %8.0g                 Was a group member on 1st July 2005
D_ID            byte                          District indicator
------------------------------------------------------------------------------------------------------------
```

Proportion of items owned

```
. gen RaschProp = RaschScore / 9
```

GLM estimation

```
.     xi:     glm         RaschProp     NoLoansFirstYearOfGroup     NoLoansFifthYearOfGroup
CVLoansFirstYearOfGroup       CVLoansFifthYearOfGroup              logPrincipCPIFirstLoan
logPrincipCPILastLoan  PovEntryPoor  PovEntryUltra  CustomerEverDelinq30pc  i.D_ID  [pw=
pwFullModel]    if  MemberOn20050701  &  NoLoansFifthYearOfGroup  <  3238  ,  link(logit)
family(binomial) vce(robust) nolog

[The condition "NoLoansFifthYearOfGroup < 3238" excludes a sample member erroneously
tagged with the ID of non-individual borrowers from the loan table.]

i.D_ID            _ID_ID_1-11         (naturally coded; _ID_ID_1 omitted)
note: _ID_ID_8 dropped because of collinearity
note: _ID_ID_9 dropped because of collinearity
note: _ID_ID_10 dropped because of collinearity
note: _ID_ID_11 dropped because of collinearity

note: RaschProp has noninteger values

Generalized linear models                         No. of obs      =      4188
Optimization     : ML                             Residual df     =      4172
                                                  Scale parameter =         1
Deviance         =  5133.459495                   (1/df) Deviance =  1.230455
Pearson          =  4772.062272                   (1/df) Pearson  =  1.143831

Variance function: V(u) = u*(1-u/1)               [Binomial]
Link function    : g(u) = ln(u/(1-u))             [Logit]

                                                  AIC             =  6.411877
```

```
Log pseudolikelihood = -13410.46949                BIC             = -29660.93

------------------------------------------------------------------------------
             |               Robust
   RaschProp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
NoLoansFir~p |  -.0002472   .0015292    -0.16   0.872    -.0032444    .0027501
NoLoansFif~p |   .0033257   .0020569     1.62   0.106    -.0007057    .0073572
CVLoansFir~p |  -.3161426   .1280355    -2.47   0.014    -.5670876   -.0651976
CVLoansFif~p |   .4229767   .0847596     4.99   0.000      .256851    .5891024
logP~rstLoan |   .6282162   .0777101     8.08   0.000     .4759072    .7805251
logP~astLoan |     .61207   .0717931     8.53   0.000     .4713581    .7527818
PovEntryPoor |  -.1294969    .050248    -2.58   0.010    -.2279812   -.0310127
PovEntryUl~a |  -.3186239   .0647945    -4.92   0.000    -.4456188   -.1916289
Custome~30pc |  -.1016436   .0415281    -2.45   0.014    -.1830372    -.02025
    _ID_ID_2 |   .1036983   .0469296     2.21   0.027     .0117179    .1956787
    _ID_ID_3 |   .0226329   .0902008     0.25   0.802    -.1541573    .1994232
    _ID_ID_4 |   .0992195   .0740237     1.34   0.180    -.0458643    .2443033
    _ID_ID_5 |   .1229181   .1038601     1.18   0.237     -.080644    .3264801
    _ID_ID_6 |  -.2389336   .0449606    -5.31   0.000    -.3270546   -.1508125
    _ID_ID_7 |  -.3434148   .0507178    -6.77   0.000    -.4428199   -.2440098
       _cons |  -5.127295   .3278458   -15.64   0.000    -5.769861   -4.484729
------------------------------------------------------------------------------
```

Adjusted predictions

Predictions for poverty groups when size of last loan left as is

```
. adjust NoLoansFirstYearOfGroup NoLoansFifthYearOfGroup CVLoansFirstYearOfGroup
CVLoansFifthYearOfGroup logPrincipCPIFirstLoan CustomerEverDelinq30pc _ID_ID_2-
_ID_ID_7 if MemberOn20050701 & NoLoansFifthYearOfGroup < 3238 , by(PovEntryPoor
PovEntryUltra ) gen( RaschProp2adj) xb

-------------------------------------------------------------------------------------------------------------------
     Dependent variable: RaschProp     Equation: RaschProp     Command: glm

       Created variable: RaschProp2xb

    Variable left as is: logP~astLoan

 Covariates set to mean: NoLoansFir~p = 21.274355, NoLoansFif~p = 17.45702, CVLoansFir~p = .40016405,
                         CVLoansFif~p = .518387, logP~rstLoan = 3.823596, Custome~30pc = .16260745, _ID_ID_2
                         = .24808978, _ID_ID_3 = .03414518, _ID_ID_4 = .06757402, _ID_ID_5 = .02172875, _ID_ID_6
                         = .25191022, _ID_ID_7 = .13037249
-------------------------------------------------------------------------------------------------------------------
```

Linear prediction

```
------------------------------
Implied   |
poverty   |  Implied poverty
level at  |  level at entry:
entry:    |     ultra-poor
poor      |        0         1
----------+-------------------
        0 | -.138144  -.539027
        1 | -.283784
------------------------------
     Key:  Linear Prediction

. summ  RaschProp2xb

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
RaschProp2xb |      4188   -.3068195    .2038581  -1.556338   .5159177
```

Predicted proportion

```
. gen  RaschProp2logit = exp(RaschProp2xb) / (1+ exp( RaschProp2xb))

. summ  RaschProp2logit

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
RaschProp2~t |      4188    .4246264    .0492642   .1741727   .6261927
```

```
. label var  RaschProp2logit "Prop HH items, adjusted prediction"
```

Predicted number of items owned

```
. gen  RaschSc2adj = 9*RaschProp2logit

. summ  RaschSc2adj

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
 RaschSc2adj |      4188    3.821638    .4433781   1.567554   5.635734

. label var RaschSc2adj "Household items, adjusted prediction"
```

Graph

See study report. Syntax:

```
.   twoway   (line   RaschSc2adj   DisbursedAmtCPIadjLastLoan   if   PovEntryPoor   ==0   &
PovEntryUltra==0,   sort   lwidth(medthick))   (line   RaschSc2adj   Disburs~astLoan   if
PovEntryPoor   ==1   &     PovEntryUltra==0   &   DisbursedAmtCPIadjLastLoan   >   500,   sort
lwidth(medthick))  (line  RaschSc2adj  DisbursedAmtCPIadjLastLoan  if  PovEntryPoor  ==0  &
PovEntryUltra==1 & DisbursedAmtCPIadjLastLoan > 500, sort lwidth(medthick)), xscale(log)
xline(500  1000  2000  5000  10000  20000  50000  100000  200000,  lwidth(vthin)  lcolor(gs14))
xlabel(500  1000  2000  5000  10000  20000  50000  100000  200000)  title(Borrower  household
wealth after 5 - 6 years in program) subtitle(Items owned from a set of nine durable
consumer goods) note("Note: 4,188 borrowers who were active on 1 July 2005 and were
surveyed between Jan 2010 and ""June 2011. - Predictions adjusted for: Districts;
borrower group activity; borrower's first loan and""delinquency history. - GLM model of
proportion of items owned.", span) legend(order(1 "Non-poor" 2 "Poor" 3 "Ultra-poor")
rows(1) title(Borrower recruited through a program primarily aimed at, size(medsmall)))
```

Branch level summary for GIS

Source:

```
log:
C:\...\Analyses\Analyses110618_3_CollapseToBranches\110618_1231AB_CollapseToBranchesCont.
log

.                                                                                    use
"C:\...\Analyses\Analyses110616_2_BorrowerSurvey\110616_1649AB_BorrowerSurvey_work09.dta


. gen byte IsWoman = (Sex == "F")
. destring  AO_ID Uz_Id Br_ID Br_U_ID, replace

.                                                                                  save
"C:\...\Analyses\Analyses110618_3_CollapseToBranches\110618_1240AB_BranchLevelSummary.dta
"

. collapse (count) recno   HasLoanData     (mean) Age IsWoman MaleAdults FemaleAdults
BoysSchoolAge    GirlsSchoolAge    BoysSmall    GirlsSmall    BoysInSchool    GirlsInSchool
BoysInCollege       GirlsInCollege       MealsEatenYesterday       MonthsFamilyGoesHungry
LitersOfCookingOilUsedPast7Days LandOwnHomesteadInDec  LandOwnFarmInDec SanitLatrineUsing
OwnCotOrBed OwnChairOrBench OwnAlmirah OwnSewingMachine OwnWatchOrClock OwnElectricit
> y OwnTV OwnTelephone OwnMotorBike YearJoinedRDRS OwnRaschScore OwnRaschLatent  Br_ID
Br_U_ID AO_ID U_id Uz_Id D_ID YearFounded  GroupIsFemale NoMembersEver LengthPartnerGroup
GrMembersOn20050701   GrMembersOn20070701   GrMembersOn20090701   NoLoansFirstYearOfGroup
NoLoansThirdYearOfGroup   NoLoansFifthYearOfGroup   NumberLoansByCustomer   LengthPartner
DisbursedAmtCPIadjFirstLoan   DaysUnderObs   DisbursedAmtCPIadjLastLoan   SumAllLoansCPIadj
MemberOn20050701      MemberOn20070701      MemberOn20090701      CustomerEverDelinq01pc
CustomerEverDelinq05pc           CustomerEverDelinq10pc           CustomerEverDelinq20pc
CustomerEverDelinq30pc CustomerEverDelinq40pc CustomerEverDelinq50pc , by(BranchName)

. save

. summ  recno HasLoanData

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       recno |       156    488.4103    55.38066          1        535
 HasLoanData |       156    488.4103    55.38066          1        535

. sort recno
```

```
. ren recno SampleSize

. drop  HasLoanData
```

So far we have mapped only the average number of durable consumer goods among the borrowers sampled in a branch area:

```
. summ  OwnRaschScore, detail

                    (mean) OwnRaschScore
-------------------------------------------------------------
      Percentiles      Smallest
 1%     2.049713          1.618
 5%     2.402985       2.049713
10%     2.693487          2.074       Obs                 156
25%     3.011141       2.244533       Sum of Wgt.         156

50%     3.500755                      Mean           3.516452
                        Largest       Std. Dev.      .6870024
75%        3.934          5.06
90%     4.359282       5.109533       Variance       .4719723
95%     4.859564       5.179641       Skewness       .1437735
99%     5.179641       5.253937       Kurtosis       2.955703
```

See map in executive summary of study report. Note that survey data was supplied from all 156 branches (with an outlier of 1 in sample sizes). However, we have loan records in the loan master table only from 151 branches.

Correspondence and permissions

Correspondence regarding the analysis should be addressed to Aldo Benini, *aldobenini [at] gmail.com.*

We will share data files with bona-fide researchers who obtain permission from RDRS Bangladesh. Applications are to be sent to:

Mr. Tapan K. Karmaker
Director, Microfinance Program
RDRS Bangladesh
Rangpur
Bangladesh

E-mail: *tapan [at] rdrsrangpur.org.*

References

Bangladesh Bank (2010) "Consumer price index and inflation rate in Bangladesh [Table VIII]." Retrieved 19 November 2010, from http://www.bangladesh-bank.org/pub/monthly/econtrds/econtrdsviii.pdf.

Baum, C. F. (2008). "Stata tip 63: Modeling proportions." The Stata Journal **8**(2): 299-303.

Benini, A. (2007). The Wealth of the Poor. Simplifying living standards measurements with Rasch scales? [unpubl. ms.]. Washington DC.

Blasnik, M. and K. Smith (Undated). nearmrg - Nearest match merging of datasets [STATA ado and help file], M Blasnik & Associates and Clinical Epidemiology and Biostatistics Unit, Murdoch Childrens Research Institute.

Burke, W. (2009). "Fitting and interpreting Cragg's tobit alternative using Stata." Stata Journal **9**(4): 584-592.

Hardouin, J.-B. (2007). "Rasch analysis: Estimation and tests with raschtest." The STATA Journal **7**(1): 22-44.

Mokken, R. J. (1971). A Theory and Procedure of Scale Analysis: With Applications in Political Research The Hague, Mouton

Version 19 September 2011 / AB