Reading IDS data into R

Göran Broström

30 november, 2016

Introduction

- Read Individual.txt and IindivIndiv.txt into **R**.
- Perform some selections and transformations.
- Save into **R** format, .rda files.

Reading the INDIVIDUAL file

```
options(stringsAsFactors = FALSE)

individual <- read.table("rawdata/Individual.txt",
                         header = FALSE)

names(individual) <- c("Id", "IdD", "IdI", "Source",
                       "Type", "Value", "ValueIdC",
                       "Day", "Month", "Year",
                       "StartDay", "StartMonth", "StartYear",
                       "EndDay", "EndMonth", "EndYear",
                       "DateOfOccurrence", "TimeInvariant")

save(individual, file = "data/individual.rda")
```

Looking at Type

```
print(sort(unique(individual$Type)))
```

```
##  [1] "ARRIVAL_FROM"          "BAPTISM_DATE"          "BAPTISM_LOCATION"
##  [4] "BIRTH_DATE"            "BIRTH_LOCATION"        "CHILDBIRTH_ASSISTANT"
##  [7] "CIVIL_STATUS"          "DEATH_DATE"            "DEATH_LOCATION"
## [10] "DEPARTURE_TO"          "END_OBSERVATION"       "FIRST_NAME"
## [13] "FUNERAL_DATE"          "FUNERAL_LOCATION"      "LAST_NAME"
## [16] "LEGITIMACY"            "MARRIAGE_DATE"         "MARRIAGE_LOCATION"
## [19] "MULTIPLE_BIRTH"        "OBSERVATION"           "OCCUPATION"
## [22] "OCCUPATION_HISCO"      "OCCUPATION_STANDARD"   "SEX"
## [25] "START_OBSERVATION"     "STILLBIRTH_DATE"       "STILLBIRTH_LOCATION"
## [28] "VACCINATION"
```

- Most of these variables can be thrown out.
- Keep: BIRTH_DATE, BIRTH_LOCATION, DEATH_DATE, END_OBSERVATION, LEGITIMACY, MULTIPLE_BIRTH, OCCUPATION, START_OBSERVATION, SEX

The R package 'dplyr'

Hadley Wickham (http://hadley.nz, https://www.rstudio.com)

```
library(dplyr)
individual <- filter(individual, Type %in% c("BIRTH_DATE", "DEATH_DATE",
                     "START_OBSERVATION", "END_OBSERVATION", "LEGITIMACY",
                     "MULTIPLE_BIRTH", "OCCUPATION_HISCO", "SEX",
                     "MARRIAGE_DATE", "BIRTH_LOCATION"))
```

- filter selects observations (rows) by a condition.
- filter(data, condition)
- x %in% y returns a logical vector of the same length as x:

```
c(1, 3, 2, 5) %in% c(1, 2)

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

TimeInvariant

```
select(individual, Type, TimeInvariant) %>% table()
```

```
##                     TimeInvariant
## Type                        Time invariant
##   BIRTH_DATE           74852              0
##   BIRTH_LOCATION           0         125839
##   DEATH_DATE           75831              0
##   END_OBSERVATION      77461              0
##   LEGITIMACY               0          75831
##   MARRIAGE_DATE        77462              0
##   MULTIPLE_BIRTH           0          75831
##   OCCUPATION_HISCO    198228              0
##   SEX                      0          75831
##   START_OBSERVATION    76372              0
```

Pipes

The symbol "%>%" is a pipe: Think of it as saying and then. It is equivalent to a two-step nested procedure:

```
table(select(individual, Type, TimeInvariant))
```

or non-nested:

```
x <- select(individual, Type, TimeInvariant)
table(x)
rm(x) # Remove the temporary variable 'x'
```

The advantages with the pipe is:

- It is easy to read (once you are familiar with the pipe concept).
- It does not clutter with temporary variables.
- It is fast.

Recode TimeInvariant

```
individual$TimeInvariant <- with(individual, Type %in%
                                   c("BIRTH_DATE", "BIRTH_LOCATION",
                                     "DEATH_DATE", "LEGITIMACY",
                                     "MULTIPLE_BIRTH", "SEX"))
select(individual, Type, TimeInvariant) %>% table()
```

```
##                       TimeInvariant
## Type                    FALSE   TRUE
##   BIRTH_DATE                0  74852
##   BIRTH_LOCATION            0 125839
##   DEATH_DATE                0  75831
##   END_OBSERVATION       77461      0
##   LEGITIMACY                0  75831
##   MARRIAGE_DATE         77462      0
##   MULTIPLE_BIRTH            0  75831
##   OCCUPATION_HISCO     198228      0
##   SEX                       0  75831
##   START_OBSERVATION     76372      0
```

Data frames person and observation

```
person <- filter(individual, TimeInvariant)
person$TimeInvariant <- NULL
observation <- filter(individual, !TimeInvariant)
observation$TimeInvariant <- NULL
```

- '!' means NOT (logical negation).
- NULL means nothing (removes): The variable TimeInvariant is not needed after the split.
- person contains time-fixed variables.
- observation contains time-varying variables.

DateOfOccurrence in person

```
with(person, table(Type, DateOfOccurrence))
```

```
##                  DateOfOccurrence
## Type                        Date of occurence
##   BIRTH_DATE                0             74852
##   BIRTH_LOCATION       125839                 0
##   DEATH_DATE                0             75831
##   LEGITIMACY            75831                 0
##   MULTIPLE_BIRTH        75831                 0
##   SEX                   75831                 0
```

The information that the birth and death dates are "Date of occurrence" is trivial. Remove it:

```
person$DateOfOccurrence <- NULL
```

DateOfOccurrence in observation

```
with(observation, table(Type, DateOfOccurrence))
```

```
##                       DateOfOccurrence
## Type                   Date of occurence
##   END_OBSERVATION                  77461
##   MARRIAGE_DATE                    77462
##   OCCUPATION_HISCO                198228
##   START_OBSERVATION                76372
```

The information that the birth and death dates are "Date of occurrence" is trivial. Remove it:

```
observation$DateOfOccurrence <- NULL
```

The person data frame

```
names(person)
```

```
##  [1] "Id"         "IdD"        "IdI"        "Source"     "Type"
##  [6] "Value"      "ValueIdC"   "Day"        "Month"      "Year"
## [11] "StartDay"   "StartMonth" "StartYear"  "EndDay"     "EndMonth"
## [16] "EndYear"
```

We only need IdI, Type, Value, Day, Month, and Year:

```
person <- select(person, IdI, Type, Value, Day, Month, Year)
select(person, Day, Month, Year) %>% summary()
```

```
##       Day             Month            Year
##  Min.   : 0.000   Min.   : 0.0   Min.   :   0.0
##  1st Qu.: 0.000   1st Qu.: 0.0   1st Qu.:   0.0
##  Median : 0.000   Median : 0.0   Median :   0.0
##  Mean   : 6.128   Mean   : 2.6   Mean   : 827.2
##  3rd Qu.:11.000   3rd Qu.: 5.0   3rd Qu.:1840.0
##  Max.   :31.000   Max.   :27.0   Max.   :1903.0
```

Bad Month?

```
filter(person, Month > 12)
```

```
##        IdI       Type Value Day Month Year
## 1 13285897 DEATH_DATE        12    27 1801
```

We guess that Day and Month has been swapped for this person.

```
filter(person, IdI == 13285897)
```

```
##        IdI            Type                        Value Day Month Year
## 1 13285897     BIRTH_DATE                                26     1 1733
## 2 13285897 BIRTH_LOCATION                SKELLEFTEÅ      26     1 1733
## 3 13285897     LEGITIMACY                  Legitimate     0     0    0
## 4 13285897 MULTIPLE_BIRTH Unknown /No multiple birth     0     0    0
## 5 13285897            SEX                       Female    0     0    0
## 6 13285897     DEATH_DATE                                12    27 1801
```

Swap back

```
(row <- which(person$Month == 27))

## [1] 300181

## Swap:
temp <- person$Month[row]
person$Month[row] <- person$Day[row]
person$Day[row] <- temp
##
rm(temp, row)
filter(person, IdI == 13285897 & Type == "DEATH_DATE")

##        IdI       Type Value Day Month Year
## 1 13285897 DEATH_DATE        27    12 1801
```

Make dates

Use Day, Month, and Year to create a Date:

```
person$date <- with(person, paste(Year, Month, Day, sep = "-")) %>%
               as.Date("%Y-%m-%d")
person$Day <- person$Month <- person$Year <- NULL
str(person)

## 'data.frame': 504015 obs. of  4 variables:
##  $ IdI  : int  11005266 11005266 11005266 11005266 11005266 11005266 1105103
##  $ Type : chr  "BIRTH_DATE" "BIRTH_LOCATION" "LEGITIMACY" "MULTIPLE_BIRTH" .
##  $ Value: chr  "" "Missing information" "Unknown" "Unknown /No multiple birt
##  $ date : Date, format: NA NA ...

filter(person, !is.na(date)) %>% head(3)

##        IdI           Type                Value       date
## 1 11404184     BIRTH_DATE                      1801-11-27
## 2 11404184 BIRTH_LOCATION Missing information 1801-11-27
## 3 19206583     BIRTH_DATE                      1778-09-02
```

Date as a Value

- For BIRTH_DATE and DEATH_DATE,
  - the date is in fact the Value.

So:

```
change <- person$Type %in% c("BIRTH_DATE", "DEATH_DATE")
person$Value[change] <- person$date[change]
rm(change) # Just a temporary variable ...
person$date <- NULL # Not needed any more
head(person)
```

```
##        IdI            Type                        Value
## 1 11005266      BIRTH_DATE                         <NA>
## 2 11005266 BIRTH_LOCATION         Missing information
## 3 11005266      LEGITIMACY                      Unknown
## 4 11005266 MULTIPLE_BIRTH Unknown /No multiple birth
## 5 11005266             SEX                      Unknown
## 6 11005266      DEATH_DATE                         <NA>
```

`person` is not tidy!

- tidy:
  - Variables in columns,
  - observations in rows.
- The column Type is variable names!

Duplicates

- There are duplicate Type == BIRTH_LOCATION.
  - How many?

```
group_by(person, IdI) %>%
    summarize(count = sum(Type == "BIRTH_LOCATION")) %>%
    select(count) %>%
    table()

## .
##     0     1     2     3     4     5
##   979 69057  4854   752   142    47
```

Remove all duplicates

Remove all duplicates of IdI + Type:

```
NROW(person)

## [1] 460000

dups <- with(person, paste(IdI, Type, sep = "")) %>%
        duplicated()
person <- filter(person, !dups)
NROW(person)

## [1] 453028
```

Make person tidy!

We use Hadley Wickham's package tidyr:

```
library(tidyr)
person <- spread(person, Type, Value)
person$BIRTH_DATE <- as.Date(as.numeric(person$BIRTH_DATE),
origin = "1970-01-01")
person$DEATH_DATE <- as.Date(as.numeric(person$DEATH_DATE),
origin = "1970-01-01")
names(person) <- tolower(names(person))
str(person)
```

```
## 'data.frame': 75831 obs. of  7 variables:
##  $ idi            : int  10000192 10000250 10000325 10000447 10000649 1000066
##  $ birth_date     : Date, format: "1875-04-15" "1785-12-06" ...
##  $ birth_location: chr  "KUSMARK" "SKELLEFTEÅ " "ATTMAR" "STORKÅGE" ...
##  $ death_date     : Date, format: "1881-01-03" "1848-03-10" ...
##  $ legitimacy     : chr  "Legitimate" "Legitimate" "Unknown" "Legitimate" ...
##  $ multiple_birth: chr  "Unknown /No multiple birth" "Unknown /No multiple b
##  $ sex            : chr  "Male" "Female" "Female" "Male" ...
```

Fix BIRTH_LOCATION

```
length(unique(person$birth_location))
```

```
## [1] 968
```

```
x <- select(person, birth_location) %>%
    table() %>%
    sort(decreasing = TRUE)
x[1:12]
```

```
## .
##          SKELLEFTEÅ  Missing information             FALMARK
##               16078                9413                1909
##             KUSMARK            STORKÅGE               BUREÅ
##                1729                1717                1436
##                BÖLE             ERSMARK            BERGSBYN
##                1257                1249                1214
##           BURTRÄSK      SKELLEFTEÅ STAD           HJOGGBÖLE
##                1147                1081                1042
```

```
person$urban <- with(person,
                     birth_location %in%
                     c("SKELLEFTEÅ", "SKELLEFTEÅ STAD"))
```

Something went wrong?

```
x[c(1, 11)]
```

```
## .
##     SKELLEFTEÅ  SKELLEFTEÅ STAD
##           16078             1081
```

```
with(person, table(urban))
```

```
## urban
## FALSE  TRUE
## 74750  1081
```

What?

Fix BIRTH_LOCATION, II

Use Hadley Wickham's package stringr:

```
library(stringr)
person$birth_location <- str_trim(person$birth_location)
person$urban <- with(person,
                     birth_location %in%
                     c("SKELLEFTEÅ", "SKELLEFTEÅ STAD"))
with(person, table(urban))

## urban
## FALSE  TRUE
## 58672 17159
```

str_trim removes whitespace from start and end of a string.

Fix legitimacy

```
with(person, table(legitimacy))

## legitimacy
## Child of betrothal       Illegitimate         Legitimate
##                104               2745              52979
##            Unknown
##              20003
```

I make the daring assumption that:

```
person$legitimacy[person$legitimacy == "Child of betrothal"] <-
    "Illegitimate"
person$legitimacy[person$legitimacy == "Unknown"] <-
    "Legitimate"
person$legitimacy <- factor(person$legitimacy) %>%
    relevel(ref = "Legitimate")
with(person, table(legitimacy))

## legitimacy
##   Legitimate Illegitimate
##        72982         2849
```

Fix multiple_birth

```
with(person, table(multiple_birth))

## multiple_birth
##                          2                          3
##                       1436                         30
## Unknown /No multiple birth
##                      74365
```

We need to make a factor of this with the labels 1, 2, 3:

```
person$multiple_birth[!(person$multiple_birth %in% c(2, 3))] <- 1
person$multiple_birth <- factor(person$multiple_birth) %>%
    relevel(ref = "1")
with(person, table(multiple_birth))

## multiple_birth
##     1     2     3
## 74365  1436    30
```

The final person data frame

```
summary(person)
```

```
##       idi              birth_date          birth_location
##  Min.   :10000192   Min.   :1699-01-08   Length:75831
##  1st Qu.:12496908   1st Qu.:1816-04-25   Class :character
##  Median :14972256   Median :1851-05-22   Mode  :character
##  Mean   :14993999   Mean   :1842-01-19
##  3rd Qu.:17497322   3rd Qu.:1875-06-20
##  Max.   :19999984   Max.   :1902-02-01
##                     NA's   :11851
##    death_date                  legitimacy    multiple_birth
##  Min.   :1755-06-07   Legitimate  :72982   1:74365
##  1st Qu.:1840-04-17   Illegitimate: 2849   2: 1436
##  Median :1867-01-29                        3:   30
##  Mean   :1861-02-18
##  3rd Qu.:1884-01-11
##  Max.   :1903-10-31
##  NA's   :53301
##      sex              urban
##  Length:75831       Mode :logical
##  Class :character   FALSE:58672
##  Mode  :character   TRUE :17159
##                     NA's :0
##
```

The observation file

Fix dates and select (extremely un-tidy):

```
observation$date <- with(observation,
                         paste(Year, Month, Day, sep = "-")) %>%
                         as.Date("%Y-%m-%d")

##
observation$startdate <- with(observation,
            paste(StartYear, StartMonth, StartDay, sep = "-")) %>%
            as.Date("%Y-%m-%d")

##
observation$enddate <- with(observation,
                paste(EndYear, EndMonth, EndDay, sep = "-")) %>%
                as.Date("%Y-%m-%d")

observation <- select(observation,
                      IdI, Type, Value, date, startdate, enddate)
```

Trim and sort

```
observation$Type <- str_trim(observation$Type)
observation$Value <- str_trim(observation$Value)
observation <- arrange(observation, IdI, date, startdate, enddate)
with(observation, table(Type))
```

```
## Type
##   END_OBSERVATION     MARRIAGE_DATE  OCCUPATION_HISCO START_OBSERVATION
##             77461             77462            198228             76372
```

```
filter(observation, Type == "START_OBSERVATION") %>%
    select(Value) %>% table()
```

```
## .
##      Arrival        Birth Start source
##         9264        55845        11263
```

```
filter(observation, Type == "END_OBSERVATION") %>%
    select(Value) %>% table()
```

```
## .
##      Death  Departure End source
##      25990      13957      37514
```

Type, Value, and new variables

- Variables:
  - married (TRUE or FALSE)
  - occupation (code, 0 = no occupation)
  - present (TRUE or FALSE)
- START_OBSERVATION $\rightarrow$ (present = TRUE)
- END_OBSERVATION $\rightarrow$ (present = FALSE)
- OCCUPATION_HISCO $\rightarrow$ occupation, start = 0
- MARRIAGE_DATE $\rightarrow$ (married = TRUE), start = FALSE

Fix present

```
where <- observation$Type == "START_OBSERVATION"
observation$Type[where] <- "present"
observation$Value[where] <- "TRUE"
##
where <- observation$Type == "END_OBSERVATION"
observation$Type[where] <- "present"
observation$Value[where] <- "FALSE"
```

```
head(observation)
```

```
##        IdI              Type Value       date startdate enddate
## 1 10000192           present  TRUE 1875-04-15      <NA>    <NA>
## 2 10000192           present FALSE 1881-01-03      <NA>    <NA>
## 3 10000192 OCCUPATION_HISCO     0       <NA>      <NA>    <NA>
## 4 10000192    MARRIAGE_DATE             <NA>      <NA>    <NA>
## 5 10000250           present  TRUE 1785-12-06      <NA>    <NA>
## 6 10000250    MARRIAGE_DATE       1815-06-25      <NA>    <NA>
```

Fix occupation and married

```
where <- observation$Type == "OCCUPATION_HISCO"
observation$Type[where] <- "occupation"
##
where <- observation$Type == "MARRIAGE_DATE"
observation$Type[where] <- "married"
observation$Value[where] <- "TRUE"
head(observation)
```

```
##        IdI       Type Value       date startdate enddate
## 1 10000192    present  TRUE 1875-04-15      <NA>    <NA>
## 2 10000192    present FALSE 1881-01-03      <NA>    <NA>
## 3 10000192 occupation     0       <NA>      <NA>    <NA>
## 4 10000192    married  TRUE       <NA>      <NA>    <NA>
## 5 10000250    present  TRUE 1785-12-06      <NA>    <NA>
## 6 10000250    married  TRUE 1815-06-25      <NA>    <NA>
```

The dates are irritating . . .

The dates

```
filter(observation, !is.na(enddate)) %>% head(2)

##        IdI       Type Value date  startdate    enddate
## 1 10000250 occupation 54020 <NA> 1815-05-21 1815-06-25
## 2 10000250 occupation 62120 <NA> 1815-05-21 1815-06-25

filter(observation, IdI == 10000250) %>% print()

##        IdI       Type Value       date  startdate    enddate
## 1 10000250    present  TRUE 1785-12-06       <NA>       <NA>
## 2 10000250    married  TRUE 1815-06-25       <NA>       <NA>
## 3 10000250    present FALSE 1848-03-10       <NA>       <NA>
## 4 10000250 occupation 54020       <NA> 1815-05-21 1815-06-25
## 5 10000250 occupation 62120       <NA> 1815-05-21 1815-06-25
## 6 10000250 occupation 54020       <NA>       <NA>       <NA>
## 7 10000250 occupation 62120       <NA>       <NA>       <NA>
## 8 10000250 occupation 54020       <NA>       <NA>       <NA>
## 9 10000250 occupation 62120       <NA>       <NA>       <NA>
```

Decision about dates

- When date is missing, replace with startdate.
- Remove `startdate` and `enddate`.
- Sort by IdI and date.

```
where <- is.na(observation$date)
observation$date[where] <- observation$startdat[where]
observation <- select(observation, IdI, Type, Value, date) %>%
    arrange(IdI, date)
filter(observation, IdI == 10000250) %>% print()
```

```
##        IdI       Type Value       date
## 1 10000250    present  TRUE 1785-12-06
## 2 10000250 occupation 54020 1815-05-21
## 3 10000250 occupation 62120 1815-05-21
## 4 10000250    married  TRUE 1815-06-25
## 5 10000250    present FALSE 1848-03-10
## 6 10000250 occupation 54020       <NA>
## 7 10000250 occupation 62120       <NA>
## 8 10000250 occupation 54020       <NA>
## 9 10000250 occupation 62120       <NA>
```

Birth date

```
names(observation) <- tolower(names(observation))
indx <- match(observation$idi, person$idi)
observation$birth_date <- person$birth_date[indx]
select(observation, date, birth_date) %>% summary()

##       date               birth_date
##  Min.   :1714-02-03   Min.   :1699-01-08
##  1st Qu.:1835-10-04   1st Qu.:1812-03-04
##  Median :1864-03-27   Median :1842-02-21
##  Mean   :1856-06-18   Mean   :1837-09-11
##  3rd Qu.:1883-03-18   3rd Qu.:1866-11-11
##  Max.   :1903-12-22   Max.   :1902-02-01
##  NA's   :276363       NA's   :56191

observation <- observation[!is.na(observation$date), ]
```

New variables

```
vars <- unique(observation$type)
observation[, vars] <- NA
filter(observation, idi == 10000250) %>%
    select(type, value, present, occupation, married) %>%
    kable()
```

type	value	present	occupation	married
present	TRUE	NA	NA	NA
occupation	54020	NA	NA	NA
occupation	62120	NA	NA	NA
married	TRUE	NA	NA	NA
present	FALSE	NA	NA	NA

1. TRUE → column `present`
2. 54020 → column `occupation`
3. 62120 → column `occupation`
4. TRUE → column `married`
5. FALSE → column `present`

Fill in new variables

```
where <- observation$type == "present"
observation$present[where] <- observation$value[where]
##
where <- observation$type == "occupation"
observation$occupation[where] <- observation$value[where]
##
where <- observation$type == "married"
observation$married[where] <- observation$value[where]
##
filter(observation, idi == 10000250) %>%
    select(type, value, present, occupation, married) %>%
    kable()
```

type	value	present	occupation	married
present	TRUE	TRUE	NA	NA
occupation	54020	NA	54020	NA
occupation	62120	NA	62120	NA
married	TRUE	NA	NA	TRUE
present	FALSE	FALSE	NA	NA

Remove type and value

And fill down present, occupation, and married.

```
observation <- mutate(observation, type = NULL, value = NULL)
observation <- observation %>% group_by(idi) %>%
    fill(present, occupation, married)
filter(observation, idi == 10000250) %>% kable()
```

idi	date	birth_date	present	occupation	married
10000250	1785-12-06	1785-12-06	TRUE	NA	NA
10000250	1815-05-21	1785-12-06	TRUE	54020	NA
10000250	1815-05-21	1785-12-06	TRUE	62120	NA
10000250	1815-06-25	1785-12-06	TRUE	62120	TRUE
10000250	1848-03-10	1785-12-06	FALSE	62120	TRUE

Suggestions:

- Start married with FALSE?
- Start occupation with 0?

Start married and occupation

```
observation$firstRec <- with(observation, !duplicated(idi))
observation$lastRec <- c(observation$firstRec[-1], TRUE)
fillin <- with(observation, firstRec & is.na(occupation))
observation$occupation[fillin] <- 0
fillin <- with(observation, firstRec & is.na(married))
observation$married[fillin] <- "FALSE"
observation$married <- as.logical(observation$married)
observation$occupation <- as.integer(observation$occupation)
observation <- observation %>%
    group_by(idi) %>%
    fill(occupation, married)
```

What does it look like now?

```
head(observation)
```

```
## Source: local data frame [6 x 8]
## Groups: idi [2]
##
##        idi       date birth_date present occupation married firstRec
##      <int>     <date>     <date>   <chr>      <int>   <lgl>    <lgl>
## 1 10000192 1875-04-15 1875-04-15    TRUE          0   FALSE     TRUE
## 2 10000192 1881-01-03 1875-04-15   FALSE          0   FALSE    FALSE
## 3 10000250 1785-12-06 1785-12-06    TRUE          0   FALSE     TRUE
## 4 10000250 1815-05-21 1785-12-06    TRUE      54020   FALSE    FALSE
## 5 10000250 1815-05-21 1785-12-06    TRUE      62120   FALSE    FALSE
## 6 10000250 1815-06-25 1785-12-06    TRUE      62120    TRUE    FALSE
## # ... with 1 more variables: lastRec <lgl>
```

Must convert present to logical:

```
observation$present <- as.logical(observation$present)
```

Introducing age, enter

```
observation$enter <-
    with(observation, round(as.numeric(date - birth_date) / 365.2425, 3))
select(observation, birth_date, date, enter) %>% head()
```

```
## Source: local data frame [6 x 4]
## Groups: idi [2]
##
##        idi birth_date       date  enter
##      <int>     <date>     <date>  <dbl>
## 1 10000192 1875-04-15 1875-04-15  0.000
## 2 10000192 1875-04-15 1881-01-03  5.722
## 3 10000250 1785-12-06 1785-12-06  0.000
## 4 10000250 1785-12-06 1815-05-21 29.452
## 5 10000250 1785-12-06 1815-05-21 29.452
## 6 10000250 1785-12-06 1815-06-25 29.547
```

Introducing exit

- On last row: exit is equal to enter.
- On non-last row: exit is equal to enter on the next row.

```
observation$date2 <- c(observation$date[-1], NA)
observation$exit <- c(observation$enter[-1], 0)
observation <- filter(observation, (!lastRec) & (exit > enter) & present)
select(observation, birth_date, date, enter, exit) %>% head()
```

```
## Source: local data frame [6 x 5]
## Groups: idi [4]
##
##        idi birth_date       date  enter   exit
##      <int>     <date>     <date>  <dbl>  <dbl>
## 1 10000192 1875-04-15 1875-04-15  0.000  5.722
## 2 10000250 1785-12-06 1785-12-06  0.000 29.452
## 3 10000250 1785-12-06 1815-05-21 29.452 29.547
## 4 10000250 1785-12-06 1815-06-25 29.547 62.257
## 5 10000325 1858-09-08 1894-07-06 35.826 37.643
## 6 10000447 1862-10-13 1862-10-13  0.000 22.747
```

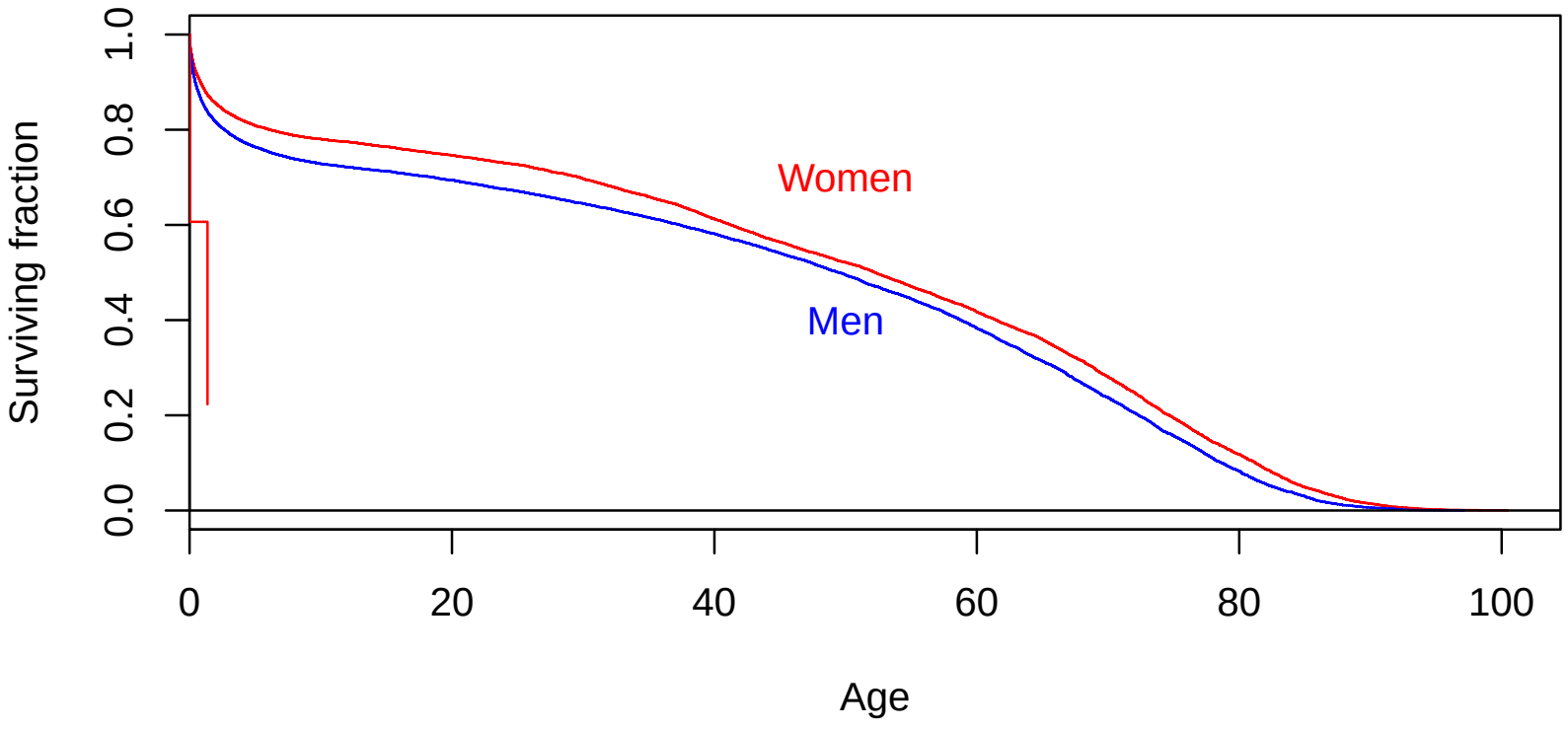
Mortality

Needs death_date from person (and some more . . .)

```
indx <- match(observation$idi, person$idi)
observation$death_date <- person$death_date[indx]
observation$sex <- person$sex[indx]
observation$event <- with(observation,
                          !is.na(death_date) &
                              !is.na(date2) &
                              death_date == date2)
with(observation, sum(event))

## [1] 18059
```

A first mortality analysis



- Kaplan-Meier estimates for women and men,
- based on 18059 deaths.

Adding info from mother

```
IndivIndiv <- read.table("rawdata/INDIV_INDIV.txt", header = FALSE)
IndivIndiv <- IndivIndiv[, c(3, 4, 6)]
IndivIndiv <- filter(IndivIndiv, V6 == "Mother")
IndivIndiv <- IndivIndiv[, 1:2]
names(IndivIndiv) <- c("mother", "child")
indx <- match(observation$idi, IndivIndiv$child) ## ERROR in handouts!!
observation$m_idi <- IndivIndiv$mother[indx]
```

Now we can put on observation from person via the key m_idi

Save . . .

```
save(observation, file = "data/observation.rda")
save(person, file = "data/person.rda")
```

See https://github.com/goranbrostrom/ACA_16/