minnesota_recidivism

Ryan Larson - UMN

August 12, 2019

```
##########################################
# Multi-State Study of Monetary Sanctions
# Ryan Larson, RA, UMN
##########################################

#Dual Debtors
  #time til first arrears - EVENT HISTORY
  #model arrears amount - fixed effects for person
    #do not drop any cases here in concerns with 1st, 2nd, etc.
    #binary in each month year of whether theres a new sentence
    #can also do severity, instead of binary, do multinomial with severity levels
      #or threshold binaries (new GM, new felony binaries)


#monsanc recidivism
  #time til next case - event history
    #start with basic survival and hazard distributions
  #running order total, running payment total, subtract columns to get debt load
  #time until payed off, what predicts 0 balance

#packages
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(tidyr)
library(stringr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

Person/Case Data

```
#processed party file
party <- read_csv("~/MonSanc/party.csv") %>%
  select(person_id, Party_Mkey, Case_Mkey, gender_impute, race_impute, name_full) %>%
  arrange(person_id, Case_Mkey)
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Case_Mkey = col_double(),
##   Party_Mkey = col_double(),
##   birth_date = col_date(format = ""),
##   address_line3 = col_logical(),
##   mnprison = col_logical(),
##   person_id = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
## Warning: 121051 parsing failures.
##  row           col           expected                 actual                   file
## 1030 address_line3 1/0/T/F/TRUE/FALSE CANADA                 '~/MonSanc/party.csv'
## 1112 address_line3 1/0/T/F/TRUE/FALSE 2305 MINNESOTA BLVD SE '~/MonSanc/party.csv'
## 1112 mnprison      1/0/T/F/TRUE/FALSE ST CLOUD               '~/MonSanc/party.csv'
## 1113 address_line3 1/0/T/F/TRUE/FALSE 2305 MINNESOTA BLVD SE '~/MonSanc/party.csv'
## 1113 mnprison      1/0/T/F/TRUE/FALSE ST CLOUD               '~/MonSanc/party.csv'
## .... ............. .................. ...................... .....................
## See problems(...) for more details.
```

```
case <- read_delim("~/MonSanc/Case.txt", delim = "|", na = "NULL",
                   col_types = list(col_character(), col_double(),
                                    col_datetime(), col_datetime(),
                                    col_character(), col_datetime(),
                                    col_character(), col_character(),
                                    col_character(), col_character(),
                                    col_character(), col_character(),
                                    col_character())) %>%
  slice(-dim(.)[1]) %>%
  select(Case_Mkey, Case_Filed_Date, Filed_County) #add as needed
```

```
## Warning: 1 parsing failure.
##      row col   expected    actual                 file
## 11188827  -- 13 columns 1 columns '~/MonSanc/Case.txt'
```

```
#merge in sentence_date via case_mkey,
  #mncis already has, vibes does not, keeping this for reference
sentence <- read_delim("~/MonSanc/Sentence.txt", delim = "|", na = "NULL", trim_ws = T) %>%
    slice(-dim(.)[1]) %>% select(Case_Mkey, Sentence_Date) %>% distinct(Case_Mkey, .keep_all = T) %>%
    mutate(Sentence_Date = as.character(format(Sentence_Date, "%Y-%m" )))
```

```
## Parsed with column specification:
## cols(
##   Charge_Mkey = col_double(),
##   Case_Mkey = col_double(),
##   Sentence_Mkey = col_double(),
```

```
##   Sentence_Type_Desc = col_character(),
##   Sentence_Date = col_datetime(format = ""),
##   Stay_Of_Imposition_Flag = col_character(),
##   Level_Of_Sentence = col_character(),
##   Level_Of_Sentence_Reduced_From_Charge_Flag = col_character(),
##   Judge_Full_Name = col_character()
## )
```

```
## Warning: 2 parsing failures.
##      row         col  expected                     actual                      file
## 10196933 Charge_Mkey a double  (10196932 row(s) affected) '~/MonSanc/Sentence.txt'
## 10196933 NA          9 columns 1 columns                  '~/MonSanc/Sentence.txt'
```

```
#person-case data with sentence date merged
person.case <- party %>% left_join(case, by = "Case_Mkey") %>%
  left_join(sentence, by = "Case_Mkey") %>%
  select(person_id, Case_Mkey, Case_Filed_Date, Sentence_Date, Filed_County, gender_impute, race_impute)
```

Financial Data

```
#MNCIS Financial - originally at transaction level
mncis <- read_delim("~/MonSanc/MNCIS_Financial.txt", delim = "|", na = "NULL",
                    col_types = list(col_double(), col_double(),
                                     col_double(), col_character(),
                                     col_character(), col_double(),
                                     col_datetime(), col_character(),
                                     col_character(), col_double(),
                                     col_double(), col_double(),
                                     col_double(), col_double(),
                                     col_double(), col_double())) %>%
  slice(-dim(.)[1])
```

```
## Warning: 2 parsing failures.
##      row       col   expected                     actual                           file
## 80480422 Case_Mkey a double   (80480421 row(s) affected) '~/MonSanc/MNCIS_Financial.txt'
## 80480422 NA        16 columns 1 columns                  '~/MonSanc/MNCIS_Financial.txt'
```

```
str(mncis)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 80480421 obs. of  16 variables:
##  $ Case_Mkey                                    : num  1399 1399 1399 1399 1399 ...
##  $ Charge_Mkey                                  : num  272792 272792 272792 272792 272792 ...
##  $ Fee_ID                                       : num  9560401 9560401 9560401 9560402 9560402 ..
##  $ Fee_Type_Desc                                : chr  "Crim/Traffic Surcharge 2003" "Crim/Traffi
##  $ Fee_Type_Category_Desc                       : chr  "Surcharges - Criminal" "Surcharges - Crim
##  $ Transaction_ID                               : num  10227691 10249547 10389607 10227691 102495
##  $ Financial_Transaction_Date                   : POSIXct, format: "2004-02-10 00:00:00" "2004-02-
##  $ Financial_Transaction_Type_Desc              : chr  "Charge" "Payment" "Disbursement" "Charge"
##  $ Financial_Transaction_Type_Detailed_Desc     : chr  NA "Mail Payment" NA NA ...
##  $ Financial_Transaction_Detail_Mkey            : num  27371979 27371974 27450582 27375244 273688
##  $ Financial_Transaction_Detail_Charge_Amount   : num  60 0 0 2 0 ...
##  $ Financial_Transaction_Detail_Credit_Amount   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Financial_Transaction_Detail_Payment_Amount  : num  0 60 0 0 2 ...
##  $ Financial_Transaction_Detail_Indirect_Amount : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Financial_Transaction_Detail_Overridden_Amount  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Financial_Transaction_Detail_Disbursement_Amount: num  0 0 60 0 0 ...
##  - attr(*, "problems")=Classes 'tbl_df', 'tbl' and 'data.frame': 2 obs. of  5 variables:
##   ..$ row     : int  80480422 80480422
##   ..$ col     : chr  "Case_Mkey" NA
##   ..$ expected: chr  "a double" "16 columns"
##   ..$ actual  : chr  "(80480421 row(s) affected)" "1 columns"
##   ..$ file    : chr  "'~/MonSanc/MNCIS_Financial.txt'" "'~/MonSanc/MNCIS_Financial.txt'"
##  - attr(*, "spec")=
##   .. cols(
##   ..   Case_Mkey = col_double(),
##   ..   Charge_Mkey = col_double(),
##   ..   Fee_ID = col_double(),
##   ..   Fee_Type_Desc = col_character(),
##   ..   Fee_Type_Category_Desc = col_character(),
##   ..   Transaction_ID = col_double(),
##   ..   Financial_Transaction_Date = col_datetime(format = ""),
##   ..   Financial_Transaction_Type_Desc = col_character(),
##   ..   Financial_Transaction_Type_Detailed_Desc = col_character(),
##   ..   Financial_Transaction_Detail_Mkey = col_double(),
##   ..   Financial_Transaction_Detail_Charge_Amount = col_double(),
##   ..   Financial_Transaction_Detail_Credit_Amount = col_double(),
##   ..   Financial_Transaction_Detail_Payment_Amount = col_double(),
##   ..   Financial_Transaction_Detail_Indirect_Amount = col_double(),
##   ..   Financial_Transaction_Detail_Overridden_Amount = col_double(),
##   ..   Financial_Transaction_Detail_Disbursement_Amount = col_double()
##   .. )
```

```
#Merging reduced category variable
mncis.bridge <- read_csv(file = "~/MonSanc/mncis_fees.csv")
```

```
## Parsed with column specification:
## cols(
##   fee_type_desc = col_character(),
##   fee_type_category_desc = col_character(),
##   new_cat = col_character()
## )
```

```
mncis$Fee_Type_Category_Desc <- str_trim(mncis$Fee_Type_Category_Desc, side = "both") #trimming wtspace
mncis$Fee_Type_Desc <- str_trim(mncis$Fee_Type_Desc, side = "both") #trimming wtspace

mncis <- mncis %>% left_join(mncis.bridge,
      by = c("Fee_Type_Desc"="fee_type_desc", "Fee_Type_Category_Desc"="fee_type_category_desc"))

#mncis time series
  #mncis financial transaction date for ordered IS the sentence date
mncis.time <- mncis %>%
  select(Case_Mkey, new_cat,
         Financial_Transaction_Detail_Charge_Amount,
         Financial_Transaction_Detail_Payment_Amount,
         Financial_Transaction_Detail_Credit_Amount,
         Financial_Transaction_Date) %>%
  rename(mncis_ordered = Financial_Transaction_Detail_Charge_Amount,
         mncis_collected = Financial_Transaction_Detail_Payment_Amount,
         mncis_credit = Financial_Transaction_Detail_Credit_Amount,
```

```
        type = new_cat,
        date = Financial_Transaction_Date) %>%
  mutate(mncis_ordered_adj = mncis_ordered-mncis_credit) %>%
  mutate(mncis_ordered_adj = ifelse(mncis_ordered_adj < 0, 0, mncis_ordered_adj),
        mncis_collected = ifelse(mncis_collected < 0, 0, mncis_collected),
       date = format(date, "%Y-%m")) %>%
  filter(type!="BAIL", type!="UNK") %>%
  select(-type, -mncis_ordered) %>%
  group_by(Case_Mkey, date) %>%
  summarise(mncis_ordered = sum(mncis_ordered_adj, na.rm = T),
            mncis_collected = sum(mncis_collected, na.rm=T)) %>%
  arrange(Case_Mkey, date)


#vibes - made to match mncis.time
#VIBES financial
vibes <- read_delim("~/MonSanc/VIBES_Financial.txt", delim = "|", na = "NULL", skip=2) %>%
  slice(-dim(.)[1])
```

```
## Parsed with column specification:
## cols(
##   Incident_ID = col_double(),
##   CNTY_CODE = col_double(),
##   Case_Filed_Date = col_datetime(format = ""),
##   Filed_Year = col_double(),
##   Imposed_Fee_ID = col_double(),
##   FEE_CODE = col_character(),
##   FEE_DESC = col_character(),
##   FEE_CATEGORY = col_character(),
##   Total_Assessments = col_double(),
##   Total_Payments = col_double(),
##   Payment_Trans_ID = col_double(),
##   Payment_Date = col_datetime(format = ""),
##   Payment_Amount = col_double()
## )
```

```
## Warning: 2 parsing failures.
##      row         col   expected                      actual                              file
## 20369988 Incident_ID a double   (20369987 row(s) affected) '~/MonSanc/VIBES_Financial.txt'
## 20369988 NA          13 columns 1 columns                  '~/MonSanc/VIBES_Financial.txt'
```

```
#vibes time series
  #vibes does not have sentence date, must merge in from sentence
vibes.clean <- vibes %>%
  select(Incident_ID,
         Total_Assessments,
         Total_Payments,
         Payment_Date) %>%
  mutate(vibes_ordered = ifelse(Total_Assessments < 0, 0, Total_Assessments),
         vibes_collected = ifelse(Total_Assessments < 0, 0, Total_Assessments),
         payment_date = format(Payment_Date, "%Y-%m")) %>%
  select(-Payment_Date, -Total_Assessments, -Total_Payments) %>%
  mutate(vibes_ordered = ifelse(is.na(vibes_ordered), 0, vibes_ordered),
         vibes_collected = ifelse(is.na(vibes_collected), 0, vibes_collected))
```

```r
#merge vibes bridge to vibes, then select unique identifiers merge
vibes.bridge <- read_delim("~/MonSanc/VIBES_Bridge.txt", delim = "|", na = "NULL") %>%
  slice(-dim(.)[1])
```

```
## Parsed with column specification:
## cols(
##   Case_Mkey = col_double(),
##   INCIDENT_ID = col_double()
## )
```

```
## Warning: 2 parsing failures.
##     row       col  expected                    actual                          file
## 4408290 Case_Mkey a double  (4408289 row(s) affected) '~/MonSanc/VIBES_Bridge.txt'
## 4408290 NA        2 columns 1 columns                 '~/MonSanc/VIBES_Bridge.txt'
```

```r
vibes.merge <- vibes %>% left_join(vibes.bridge, by = c("Incident_ID" = "INCIDENT_ID")) %>%
  mutate(county = ifelse(CNTY_CODE==62, "Ramsey County", "Hennepin County")) %>%
  select(Case_Mkey, county, Incident_ID) %>% distinct(Incident_ID, .keep_all = T)

vibes.clean <- vibes.clean %>% left_join(vibes.merge, by = "Incident_ID")

#merge in sentence date from person.case object (have to merge by Case_Mkey and county)
vibes.clean <- vibes.clean %>% left_join(person.case, by = c("county"="Filed_County", "Case_Mkey")) %>%
  select(Incident_ID, Case_Mkey, vibes_ordered, vibes_collected, Sentence_Date, payment_date)


vibes.ordered <- vibes.clean %>%
  select(Incident_ID, vibes_ordered, Sentence_Date) %>%
  rename(date = Sentence_Date)
vibes.payed <- vibes.clean %>% select(Incident_ID, vibes_collected, payment_date) %>%
  rename(date = payment_date)

vibes.time <- vibes.ordered %>% bind_rows(vibes.payed) %>%
    group_by(Incident_ID, date) %>%
    summarize(vibes_ordered = sum(vibes_ordered, na.rm=T),
              vibes_collected = sum(vibes_collected, na.rm=T)) %>%
    arrange(Incident_ID, date) %>%
    filter(!is.na(date)) #need to take a look at this

vibes.time <- vibes.time %>% left_join(vibes.merge, by = "Incident_ID")

#merging person.case to mncis and vibes series (have to do merges separately before appending due to id
mncis.person <- mncis.time %>% left_join(person.case, by = "Case_Mkey") %>%
  rename(ordered = mncis_ordered, collected = mncis_collected, county = Filed_County)
vibes.person <- vibes.time %>% left_join(person.case, by = c("county"="Filed_County", "Case_Mkey")) %>%
  rename(ordered = vibes_ordered, collected = vibes_collected) %>% ungroup() %>% select(-Incident_ID)

#combining
fin.series <- mncis.person %>% bind_rows(vibes.person) %>% arrange(person_id, date, Case_Mkey)

#create case count (data already arranged by date)
count <- fin.series %>% select(person_id, Case_Mkey) %>% distinct(person_id,Case_Mkey) %>%
  group_by(person_id) %>% mutate(count = row_number())

fin.series <- fin.series %>% left_join(count, by = c("person_id", "Case_Mkey"))
```

```
rm(list = ls()[!ls() %in% c("fin.series")])
```

Event History Framework - Time Until Second Case

```
#keep only first case per person ID and date of second case if one exists
#create dataset of just first date of second case
event <- fin.series %>% filter(count==2) %>% select(person_id, Case_Mkey, date) %>% distinct(Case_Mkey,

#reduce series to just first offense
eha <- fin.series %>% filter(count==1)

#merge event case_mkey and date back to series
eha <- eha %>% left_join(event, by = "person_id") %>% select(-count)

#CLOCK
#analysis period start: 2010-01-01
#clock start: date (sentencing date)
#clock end: 2015-12-31

#mutate date var and drop pre-2010 series (vibes missings)
#fill in event dates
#create binary event indicator
#alter event indicator and end date to reflect end of clock
#drop spells (e.g. payments after recid.) made after end_date
eha <- eha %>%
  mutate(date = as.Date(paste(date, 01, sep="-"), format = "%Y-%m-%d"),
         event_date = as.Date(paste(event_date, 01, sep="-"), format = "%Y-%m-%d"),
         sentence_date = as.Date(paste(Sentence_Date, 01, sep="-"), format = "%Y-%m-%d")) %>%
  select(-Sentence_Date) %>%
  filter(date >= "2010-01-01") %>%
  mutate(event = if_else(is.na(event_date)==T, 0, 1), #event indicator
         end_date = if_else(is.na(event_date)==T, as.Date("2015-12-31"), event_date), #fill censored da
         event = if_else(end_date > as.Date("2015-12-31"), 0, event), #censor cases that recidivate aft
         end_date = if_else(end_date > as.Date("2015-12-31"), as.Date("2015-12-31"), end_date)) %>%
  filter(date <= end_date) #keep cases before end_date

#expand series by month, , fill NAs, remove spell if a) before case start, or b) after end_date
eha <- eha %>% group_by(person_id) %>%
  complete(date = seq.Date(min(date), max(end_date), by = "month")) %>%
  fill(Case_Mkey, Case_Filed_Date, sentence_date, county, event_case_mkey, event_date, end_date) %>%
  filter(date >= sentence_date & date <= end_date) %>%
  mutate(ordered = replace_na(ordered, 0), collected = replace_na(collected, 0),
         event = replace_na(event, 0)) %>% group_by(person_id) %>%
  mutate(event = ifelse(sum(event, na.rm = T)>0 & date==event_date, 1, 0))

#create running vars
eha <- eha %>% group_by(person_id) %>%
  mutate(cum_order = cumsum(ordered),
         cum_coll  = cumsum(collected)) %>%
  ungroup() %>%
  mutate(lfo_debt = cum_order-cum_coll) %>%
```

```
  filter(lfo_debt >= 0) %>% #check on this at some point
  mutate(interval_date = date %m+% months(1)-days(1),
         enter = as.numeric(date)-as.numeric(sentence_date),
         exit = as.numeric(interval_date)-as.numeric(sentence_date))
```

```
library(survminer)
```

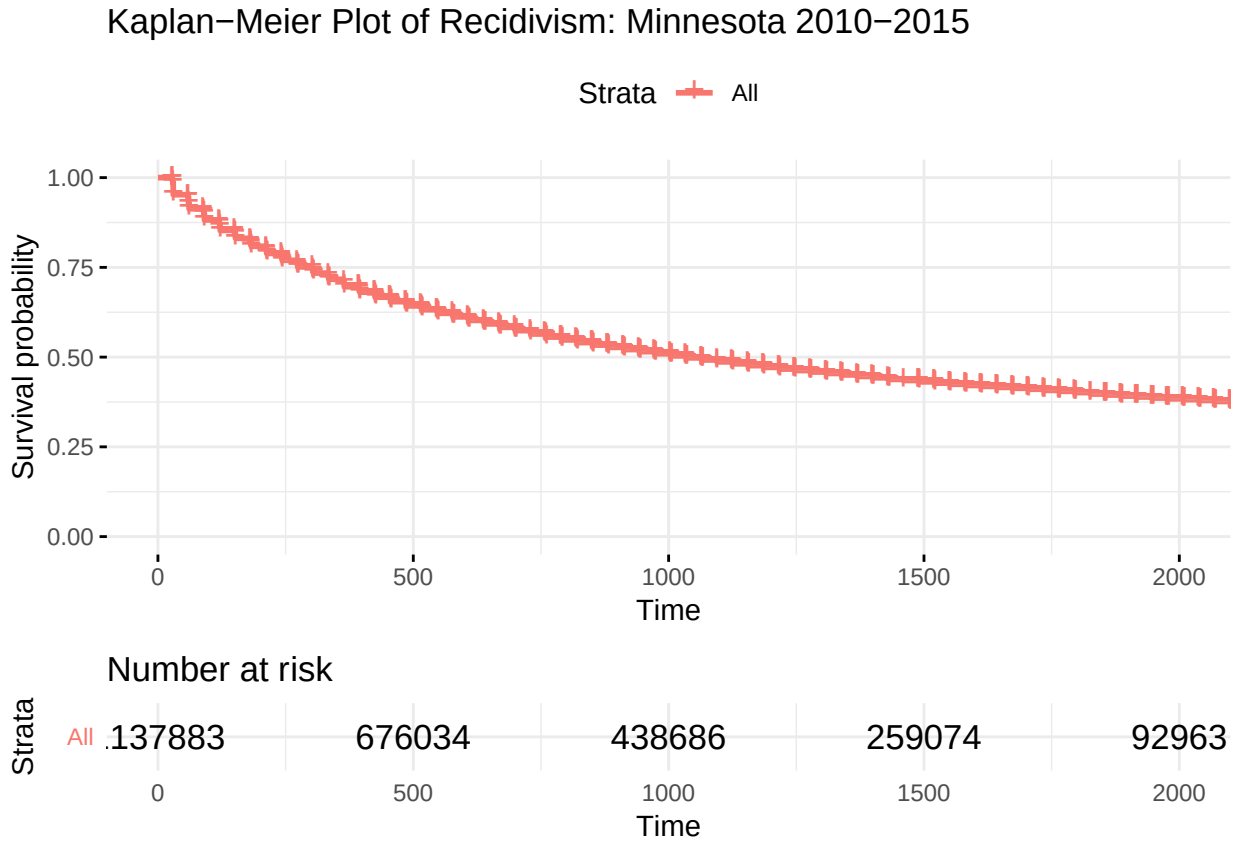
```
## Loading required package: ggpubr
```

```
## Loading required package: magrittr
```

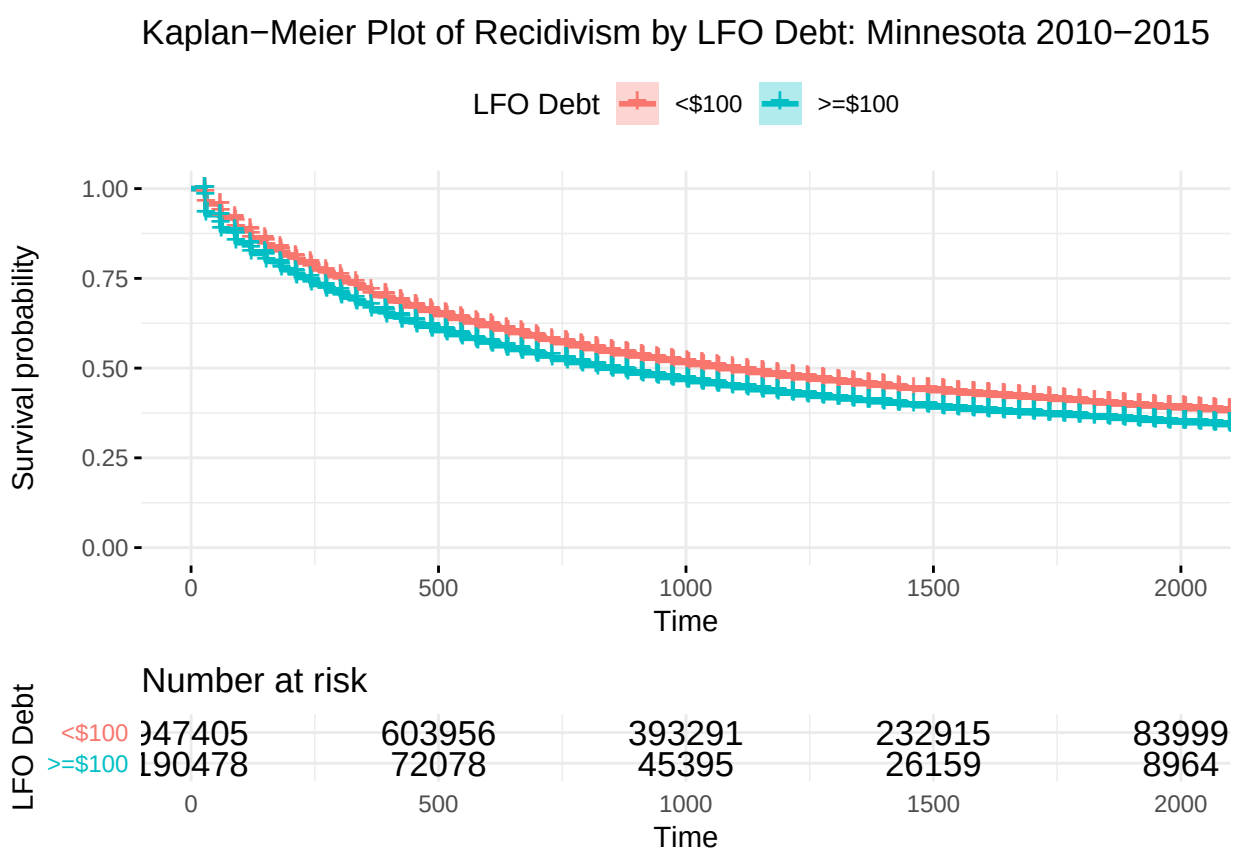
```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

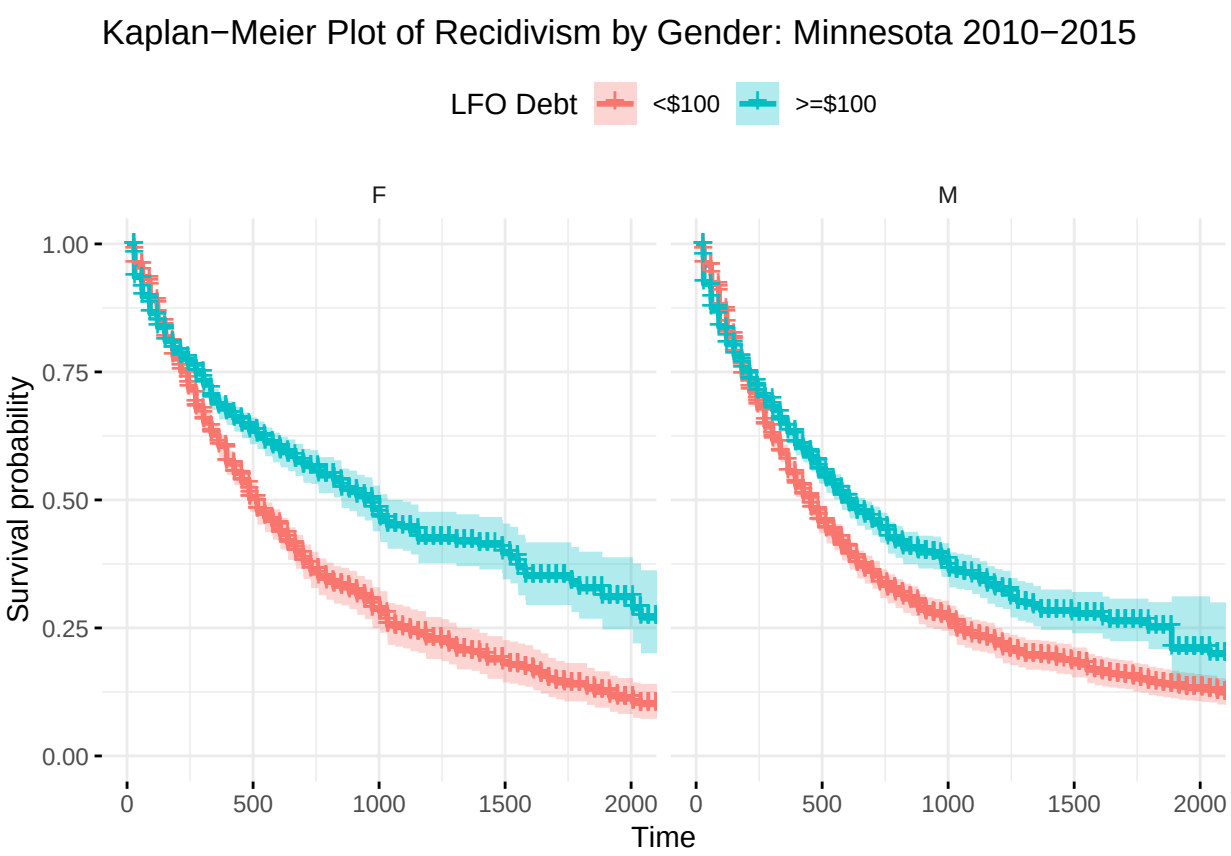
```
library(survival)

fit <- survfit(Surv(time = enter, time2 = exit, event = event) ~ 1, data = eha)

ggsurvplot(fit,
           risk.table = T,
           conf.int = T,
           xlim = c(0,2000),
           break.time.by = 500,
           ggtheme = theme_minimal(),
           title = "Kaplan-Meier Plot of Recidivism: Minnesota 2010-2015")
```



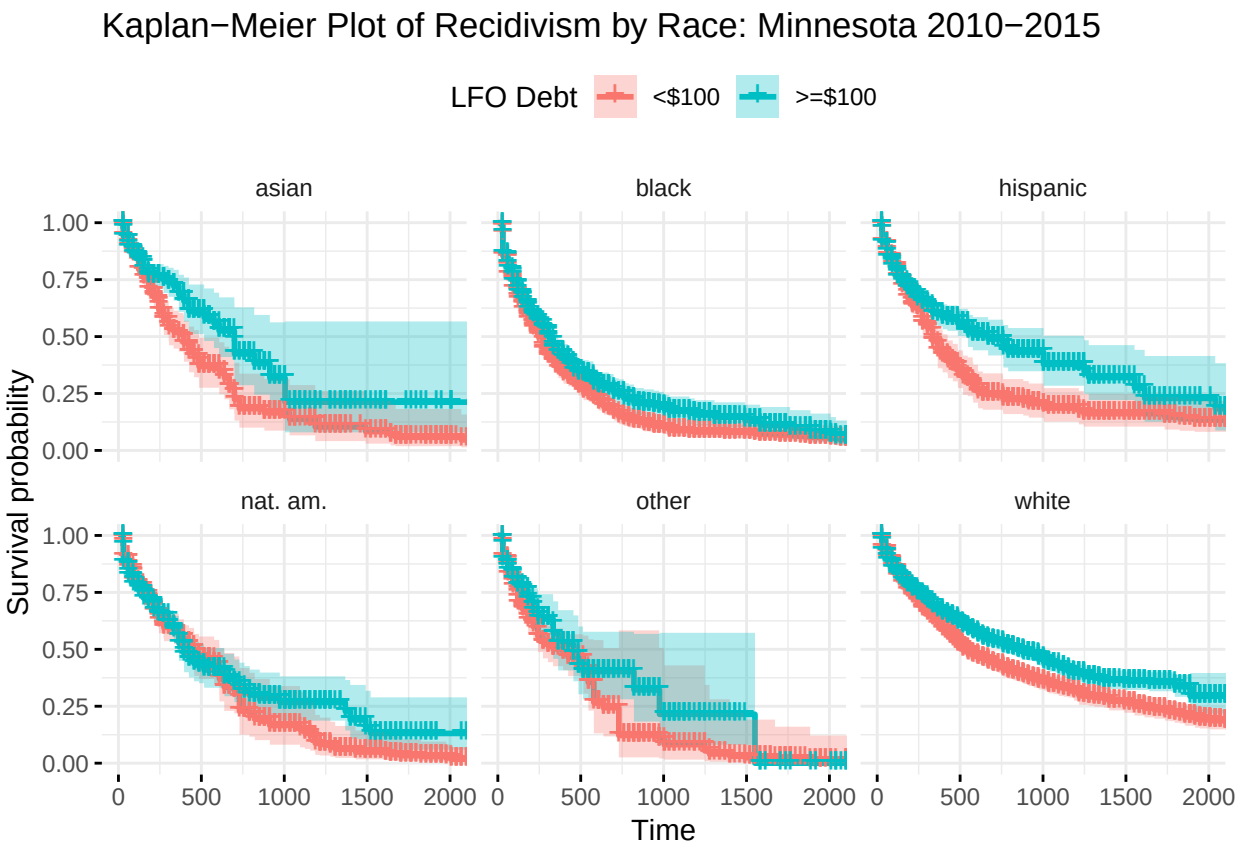

```
#just to get a preliminary look at debt
eha <- eha %>% mutate(lfo_bin = ifelse(lfo_debt >= 100, 1, 0))

fit2 <- survfit(Surv(time = enter, time2 = exit,event = event) ~ lfo_bin, data = eha)

ggsurvplot(fit2,
           risk.table = T,
           conf.int = T,
           xlim = c(0,2000),
           break.time.by = 500,
           pval=F,
           ggtheme = theme_minimal(),
           title = "Kaplan-Meier Plot of Recidivism by LFO Debt: Minnesota 2010-2015",
           legend.title = "LFO Debt",
           legend.labs = c("<$100", ">=$100"))
```




```
ggsurvplot_facet(fit2, eha, facet.by = "gender_impute",  conf.int = T,
           xlim = c(0,2000),
           break.time.by = 500,
           ggtheme = theme_minimal(),
           title = "Kaplan-Meier Plot of Recidivism by Gender: Minnesota 2010-2015",
           legend.title = "LFO Debt",
           legend.labs = c("<$100", ">=$100"),
           short.panel.labs = T)
```




```
ggsurvplot_facet(fit2, eha, facet.by = "race_impute",  conf.int = T,
           xlim = c(0,2000),
           break.time.by = 500,
           ggtheme = theme_minimal(),
           title = "Kaplan-Meier Plot of Recidivism by Race: Minnesota 2010-2015",
           legend.title = "LFO Debt",
           legend.labs = c("<$100", ">=$100"),
           short.panel.labs = T)
```



Fixed Effects Framework