

Model B : Estimation Results

Contents

Load environmet	1
Load data	2
Remove cases	2
Categorize covariates	5
Age diagnostic	5
Estimation prep	6
Specifications	8
Fitting function	8
Model	9
msm options	10
elect options	10
Model results	11
age	11
age at baseline	14
male	17
education	20
Session Info	23
Estimation results of Model B .	

Load environmet

```
base::source("http://www.ucl.ac.uk/~ucakadl/ELECT/ELECT.r") # load ELECT functions
base::source("./scripts/ELECT-utility-functions.R") # ELECT utility functions
```

```
library(magrittr) #Pipes
library(msm)
requireNamespace("ggplot2", quietly=TRUE)
requireNamespace("dplyr", quietly=TRUE)
requireNamespace("testit", quietly=TRUE)
```

Save fitted models here :

```
[1] "./data/shared/derived/models/model-b/"
```

Load data

```
# first, the script `0-ellis-island.R` imports and cleans the raw data
# second, the script `1-encode-multistate.R` augments the data with multi-states
# load this data transfer object (dto)
dto <- readRDS("./data/unshared/derived/dto.rds")

names(dto)
names(dto[["unitData"]])      # 1st element - unit(person) level data
names(dto[["metaData"]])      # 2nd element - meta data, info about variables
names(dto[["ms_mmse"]])        # 3rd element - data for MMSE outcome
ds_miss <- dto$ms_mmse$missing # data after encoding missing states (-1, -2)
ds_ms <- dto$ms_mmse$multi     # data after encoding multistates (1,2,3,4)

# compare before and after ms encoding
view_id <- function(ds1,ds2,id){
  cat("Before ms encoding:", "\n")
  print(ds1[ds1$id==id,])
  cat("\nAfter ms encoding", "\n")
  print(ds2[ds2$id==id,])
}
ids <- sample(unique(ds_miss$id),1) # view a random person for sporadic inspections
# ids <- c(75507759) #, 37125649, 50101073, 6804844, 83001827 , 56751351, 13485298, 56751351, 75507759)
view_id(ds_miss, ds_ms, ids)
```

Before ms encoding:

	id	fu_year	died	age_bl	male	edu	age_death	age_at_visit	mmse	presumed_alive
6318	59166782	0	0	83.8412	FALSE	16	NA	83.84120	29	TRUE
6319	59166782	1	0	83.8412	FALSE	16	NA	84.84052	30	TRUE
6320	59166782	2	0	83.8412	FALSE	16	NA	85.81793	29	TRUE
6321	59166782	3	0	83.8412	FALSE	16	NA	86.84736	-2	TRUE

After ms encoding

	id	fu_year	died	age_bl	male	edu	age	state	presumed_alive	mmse	firstobs
6318	59166782	0	0	83.8412	FALSE	16	83.84120	1	TRUE	29	1
6319	59166782	1	0	83.8412	FALSE	16	84.84052	1	TRUE	30	0
6320	59166782	2	0	83.8412	FALSE	16	85.81793	1	TRUE	29	0
6321	59166782	3	0	83.8412	FALSE	16	86.84736	-2	TRUE	-2	0

Remove cases

```
## REMOVE observations with missing age
cat("\n Number of observations with missing age :", sum(is.na(ds_ms$age)), "\n")
```

Number of observations with missing age : 1

```
ds_clean <- ds_ms %>%
  dplyr::filter(!is.na(age))
cat("\n Number of observations with missing age :", sum(is.na(ds_clean$age)), "\n")
```

Number of observations with missing age : 0

```
## REMOVE subjects with only ONE observed data point
cat("\n Count how many subjects have *n* observed data points")
```

Count how many subjects have *n* observed data points

```
# ds_miss %>%
ds_clean %>%
  dplyr::group_by(id) %>%
  dplyr::summarize(n_data_points = n()) %>%
  dplyr::group_by(n_data_points) %>%
  dplyr::summarize(n_people=n()) %>%
  print()
```

```
# A tibble: 17 × 2
  n_data_points n_people
      <int>      <int>
1             1        119
2             2        205
3             3        184
4             4        180
5             5        190
6             6        104
7             7        108
8             8        113
9             9        127
10            10        116
11            11        110
12            12         71
13            13         21
14            14         14
15            15         13
16            16         17
17            17          3
```

```
# Determine which ids have only a single observation
remove_ids <- ds_clean %>%
  dplyr::group_by(id) %>%
  dplyr::summarize(n_data_points = n()) %>%
  dplyr::arrange(n_data_points) %>%
  dplyr::filter(n_data_points==1) %>%
  dplyr::select(id)
remove_ids <- remove_ids$id
cat("\n The number of subjects to be removed :",length(remove_ids) )
```

The number of subjects to be removed : 119

```
ds_clean <- ds_clean %>%
  dplyr::filter(!(id %in% remove_ids))
cat("\n Count how many subjects have *n* observed data points")
```

Count how many subjects have *n* observed data points

```
# ds_miss %>%
ds_clean %>%
  dplyr::group_by(id) %>%
  dplyr::summarize(n_data_points = n()) %>%
  dplyr::group_by(n_data_points) %>%
  dplyr::summarize(n_people=n()) %>%
  print()
```

```
# A tibble: 16 × 2
  n_data_points n_people
      <int>      <int>
1           2        205
2           3        184
3           4        180
4           5        190
5           6        104
6           7        108
7           8        113
8           9        127
9          10        116
10          11        110
11          12         71
12          13         21
13          14         14
14          15         13
15          16         17
16          17          3
```

```
## REMOVE subjects with IMS at firstobs = TRUE
cat("\n View subjects with intermediate missing state at first observation")
```

View subjects with intermediate missing state at first observation

```
ids_firstobs_ims <- ds_clean %>%
  dplyr::filter(firstobs == TRUE & state == -1) %>%
  dplyr::select(id) %>% print()
```

```
      id
1 80333458
2 90214403
3 90447310
4 91804757
```

```
ids_firstobs_ims <- ids_firstobs_ims[,"id"]
ds_clean <- ds_clean %>%
  dplyr::filter(!id %in% ids_firstobs_ims)
cat("\n View subjects with intermediate missing state at first observation")
```

View subjects with intermediate missing state at first observation

```
ds_clean %>%
  dplyr::filter(firstobs == TRUE & state == -1) %>%
  dplyr::select(id) %>% print()
```

```
[1] id
```

```
<0 rows> (or 0-length row.names)
```

Categorize covariates

```
ds_clean$educat <- car::Recode(ds_clean$edu,
                              " 1:9   = '-1';
                              10:11 = '0';
                              12:30 = '1';
                              ")
ds_clean$educatF <- factor(
  ds_clean$educat,
  levels = c(-1,          0,          1),
  labels = c("0-9 years", "10-11 years", ">11 years"))
# save clean data object for records and faster access
saveRDS(ds_clean, "./data/unshared/ds_clean.rds")
```

Age diagnostic

```
# Time intervals in data:
# the age difference between timepoint for each individual
intervals <- matrix(NA,nrow(ds_clean),2)
for(i in 2:nrow(ds_clean)){
  if(ds_clean$id[i]==ds_clean$id[i-1]){
    intervals[i,1] <- ds_clean$id[i]
    intervals[i,2] <- ds_clean$age[i]-ds_clean$age[i-1]
  }
  intervals <- as.data.frame(intervals)
  colnames(intervals) <- c("id", "interval")
}
cat("\n Minimum interval length : ",min(intervals, na.rm=T))
```

```
Minimum interval length : 0.00273785
```

```
cat("\n Maximum interval length : ", max(intervals, na.rm=T))
```

```
Maximum interval length : 99982430
```

```
# the age difference between timepoint for each individual
intervals <- intervals[!is.na(intervals[,2]),] # Remove NAs:
cat("\nTime intervals between observations within individuals:\n")
```

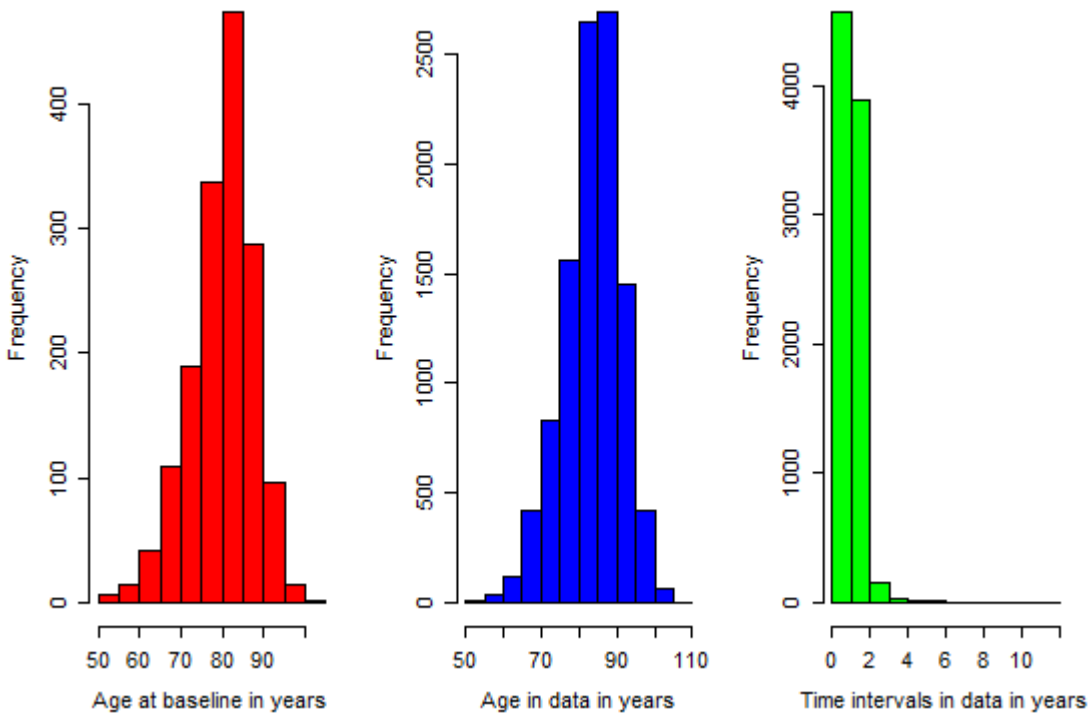
```
Time intervals between observations within individuals:
```

```
print(round(quantile(intervals[,2]),digits))
```

```
0%   25%   50%   75%  100%
0.00  0.96  1.00  1.03 11.87
```

```
# Info on age and time between observations:
opar<-par(mfrow=c(1,3), mex=0.8,mar=c(5,5,3,1))
```

```
hist(ds_clean$age[ds_clean$firstobs==1],col="red",xlab="Age at baseline in years",main="")
hist(ds_clean$age,col="blue",xlab="Age in data in years",main="")
hist(intervals[,2],col="green",xlab="Time intervals in data in years",main="")
```



```
opar<-par(mfrow=c(1,1), mex=0.8,mar=c(5,5,2,1))
```

Estimation prep

```
# list ids with intermediate missing (im) or right censored (rc) states
ids_with_im    <- unique(ds_clean[ds_clean$state == -1, "id"])
cat("\n Number of subjects with intermediate missing state(s) : ",length(ids_with_im) )
```

Number of subjects with intermediate missing state(s) : 104

```
ids_with_rc    <- unique(ds_clean[ds_clean$state == -2, "id"])
cat("\n Number of subjects with right censored state(s) : ",length(ids_with_rc) )
```

Number of subjects with right censored state(s) : 46

```
ids_with_either <- unique(c(ids_with_im, ids_with_rc))
cat("\n Number of subjects with either IMS or RC state(s) : ",length(ids_with_either) )
```

Number of subjects with either IMS or RC state(s) : 149

```
ids_with_both <- dplyr::intersect(ids_with_im, ids_with_rc)
cat("\n Number of subjects with both IMS and RC state(s) : ",length(ids_with_both) )
```

```
Number of subjects with both IMS and RC state(s) : 1
# subset a random sample of individuals if needed
set.seed(42)
ids <- sample(unique(ds_clean$id), 100)
# define the data object to be passed to the estimation call
ds <- ds_clean %>%
  # dplyr::filter(id %in% ids) %>% # make sample smaller if needed
  # exclude individuals with missing states
  # dplyr::filter(!id %in% ids_with_im) %>%
  # dplyr::filter(!id %in% ids_with_rc) %>%
  dplyr::mutate(
    male = as.numeric(male),
    age = (age - 75), # centering
    age_bl = (age_bl - 75) # centering
  ) %>%
  dplyr::select(id, age_bl,male, edu, educat, educatF,firstobs, fu_year, age, state)
# view data object to be passed to the estimation call
set.seed(42)
ids <- sample(unique(ds$id), 1)
ds %>% dplyr::filter(id %in% ids)
```

	id	age_bl	male	edu	educat	educatF	firstobs	fu_year	age	state
1	90544686	7.696783	0	12	1	>11 years	1	0	7.696783	1
2	90544686	7.696783	0	12	1	>11 years	0	1	8.682409	1
3	90544686	7.696783	0	12	1	>11 years	0	2	9.731006	1
4	90544686	7.696783	0	12	1	>11 years	0	3	10.689254	1
5	90544686	7.696783	0	12	1	>11 years	0	4	11.691307	1
6	90544686	7.696783	0	12	1	>11 years	0	5	12.709788	1
7	90544686	7.696783	0	12	1	>11 years	0	6	13.665298	1
8	90544686	7.696783	0	12	1	>11 years	0	7	14.678303	2
9	90544686	7.696783	0	12	1	>11 years	0	8	15.680356	1
10	90544686	7.696783	0	12	1	>11 years	0	9	16.709788	1

```
cat("\n Subject count : ",length(unique(ds$id)),"\n")
```

```
Subject count : 1572
```

```
cat("\n Frequency of states at baseline\n")
```

```
Frequency of states at baseline
```

```
sf <- ds %>%
  dplyr::filter(firstobs==TRUE) %>%
  dplyr::group_by(state) %>%
  dplyr::summarize(count = n()) %>% # basic frequenciies
  dplyr::mutate(pct = round(count/sum(count),2)) %>% # percentages, use for starter values
  print()
```

```
# A tibble: 3 × 3
  state count  pct
```

```

      <dbl> <int> <dbl>
1         1  1189  0.76
2         2   281  0.18
3         3   102  0.06

```

```
cat("\nState table: \n")
```

State table:

```
print(msm::statetable.msm(state,id,data=ds)) # transition frequencies
```

```

      to
from  -2  -1   1   2   3   4
-2    32   0   0   0   0   0
-1    0  25  27  13  26  47
1     32  59 4855  715 120 251
2      8  20  534  478 256 146
3      6  34   24   96 649 232

```

these will be passed as starting values

```

initial_probabilities <- as.numeric(as.data.frame(sf[!sf$state %in% c(-1,-2),"pct"])$pct)
initial_probabilities <- c(initial_probabilities,0) # no death state at first observation
cat('\n The initial values for estimation : ', initial_probabilities)

```

The initial values for estimation : 0.76 0.18 0.06 0

save the object to be used during estimation

```
saveRDS(ds, "../data/unshared/ds_estimation.rds")
```

Specifications

Fitting function

```

estimate_multistate <- function(
  model_name
  ,ds                      # data object
  ,Q                      # Q-matrix of transitions
  ,E                      # misspecification matrix
  ,qnames                 # names of the rows in the Q matrix
  ,cov_names              # string with covariate names
){
  covariates_ <- as.formula(paste0("~",cov_names))
  model <- msm(
    formula      = state ~ age,
    subject      = id,
    data         = ds,
    center       = FALSE,
    qmatrix      = Q,
    ematrix      = E,
    death        = TRUE,
    covariates    = covariates_,
    censor       = c(-1,-2),

```



```

    censor.states = list(c(1,2,3), c(1,2,3)),
    method        = method_,
    constraint     = constraint_,
    fixedpars      = fixedpars_,
    initprobs      = initprobs_,
    est.initprobs  = TRUE,
    control        = list(trace=0,REPORT=1,maxit=1000,fnscale=10000)
  )
  # model <- paste0("test", covariates_)
  saveRDS(model, paste0(pathSaveFolder,model_name,".rds"))
  return(model)
}

msm_summary <- function(model){
  cat("\n-2loglik =", model$minus2loglik,"\n")
  cat("Convergence code =", model$opt$convergence,"\n")
  p      <- model$opt$par
  p.se   <- sqrt(diag(solve(1/2*model$opt$hessian)))
  print(cbind(p=round(p,digits),
              se=round(p.se,digits),"Wald ChiSq"=round((p/p.se)^2,digits),
              "Pr>ChiSq"=round(1-pchisq((p/p.se)^2,df=1),digits)),
        quote=FALSE)
}

```

Model

```

q <- .01
# transition matrix
Q <- rbind( c(0, q, 0, q),
            c(q, 0, q, q),
            c(0, 0, 0, q),
            c(0, 0, 0, 0))
# misclassification matrix
E <- rbind( c( 0, 0, 0, 0),
            c( 0, 0, .1, 0),
            c( 0, 0, 0, 0),
            c( 0, 0, 0, 0) )
# transition names
qnames = c(
  "Healthy - Mild",    # q12
  # "Healthy - Severe", # q13
  "Healthy - Dead",    # q14
  "Mild - Healthy",    # q21
  "Mild - Severe",     # q23
  "Mild - Dead",       # q24
  # "Severe - Healthy", # q31
  # "Severe - Mild",    # q32
  "Severe - Dead"     # q34
)

```

msm options

```
digits = 2
method_ = "BFGS"      # alternatively, if does not converge "Nedler-Mead"
constraint_ = NULL     # additional model constraints
fixedpars_ = NULL      # fixed parameters
initprobs_ = initial_probabilities

# turn this chunk OFF when printing the report
# compile model objects with msm() call
# each model will be saved in the specified folder, namely pathSaveFolder
# estimate_multistate("mA1", ds, Q, E, qnames, cov_names = "age")
# estimate_multistate("mA2", ds, Q, E, qnames, cov_names = "age + age_bl")
# estimate_multistate("mA3", ds, Q, E, qnames, cov_names = "age + age_bl + male")
# estimate_multistate("mA4", ds, Q, E, qnames, cov_names = "age + age_bl + male + educat")
# estimate_multistate("mA5", ds, Q, E, qnames, cov_names = "age + age_bl + male + edu")

# assemble the list object with the results of msm estimation
models <- list()
models[["age"]][["msm"]] <- readRDS(paste0(pathSaveFolder, 'mA1.rds'))
models[["age_bl"]][["msm"]] <- readRDS(paste0(pathSaveFolder, 'mA2.rds'))
models[["male"]][["msm"]] <- readRDS(paste0(pathSaveFolder, 'mA3.rds'))
models[["educat"]][["msm"]] <- readRDS(paste0(pathSaveFolder, 'mA4.rds'))
models[["edu"]][["msm"]] <- readRDS(paste0(pathSaveFolder, 'mA5.rds'))
```

elect options

```
alive_states <- c(1,2,3)
ds_alive <- ds[ds$state %in% alive_states,]
fixedpars <- fixedpars_
age_min <- 0
age_max <- 35
age_bl <- 0
male <- 0
educat <- 0
edu <- 11

replication_n <- 1000
time_scale <- "years"
grid_par <- .5

# turn this chunk OFF when printing the report
# for(model_ in names(models) ){
#   # determine covariate list
#   if(model_=="age"){covar_list = list(age=age_min)}
#   if(model_=="age_bl"){covar_list = list(age=age_min, age_bl=age_bl)}
#   if(model_=="male"){covar_list = list(age=age_min, age_bl=age_bl, male=male)}
#   if(model_=="educat"){covar_list = list(age=age_min, age_bl=age_bl, male=male, educat=educat)}
#   if(model_=="edu"){covar_list = list(age=age_min, age_bl=age_bl, male=male, edu=edu)}
#   # compute LE
#   models[[model_]][["LE"]] <- elect(
#     model = models[[model_]][["msm"]], # fitted msm model
#     b.covariates = covar_list, # list with specified covarites values
```

```

#   statedistdata = ds_alive, # data for distribution of living states
#   time.scale.msm = time_scale, # time scale in multi-state model ("years", ...)
#   h             = grid_par, # grid parameter for integration
#   age.max       = age_max, # assumed maximum age in years
#   S             = replication_n # number of simulation cycles
# )
# # models[[model_]]["LE"] <- models[["msm"]][[model_]]
# }
# #save models estimated by elect() in a external object for faster access in the future
# saveRDS(models, paste0(pathSaveFolder,"models.rds"))

models <- readRDS(paste0(pathSaveFolder,"models.rds"))
# inspect created object
lapply(models, names)

```

```

$age
[1] "msm" "LE"

```

```

$age_bl
[1] "msm" "LE"

```

```

$male
[1] "msm" "LE"

```

```

$educat
[1] "msm" "LE"

```

```

$edu
[1] "msm" "LE"

```

Model results

age

summary

```

model <- models[["age"]]
msm_summary(model$msm)

```

```

-2loglik = 15059.61
Convergence code = 0

```

	p	se	Wald	ChiSq	Pr>ChiSq
qbase	-2.12	0.07	865.97		0.00
qbase	-4.13	0.17	558.14		0.00
qbase	-0.29	0.08	12.28		0.00
qbase	-2.19	0.15	199.54		0.00
qbase	-3.06	0.35	78.06		0.00
qbase	-2.12	0.18	134.20		0.00
qcov	0.08	0.01	160.84		0.00
qcov	0.08	0.01	31.93		0.00
qcov	-0.02	0.01	7.44		0.01

```

qcov  0.05 0.01      23.01      0.00
qcov  0.06 0.02       6.34      0.01
qcov  0.07 0.01      33.05      0.00
p      -2.15 0.09     589.75      0.00
initp -1.33 0.06     433.56      0.00
initp -2.85 0.13     480.86      0.00

```

solution

```
print(model$msm, showEnv= F)
```

Call:

```
msm(formula = state ~ age, subject = id, data = ds, qmatrix = Q,      ematrix = E, covariates = covariates)
```

Maximum likelihood estimates

Baselines are with covariates set to 0

Transition intensities with hazard ratios for each covariate

	Baseline	age
State 1 - State 1	-0.13609 (-0.15443,-0.11992)	
State 1 - State 2	0.11996 (0.10416, 0.13816)	1.0814 (1.0684,1.0946)
State 1 - State 4	0.01613 (0.01145, 0.02271)	1.0869 (1.0559,1.1187)
State 2 - State 1	0.74949 (0.63785, 0.88065)	0.9802 (0.9662,0.9944)
State 2 - State 2	-0.90846 (-1.04359,-0.79082)	
State 2 - State 3	0.11212 (0.08276, 0.15189)	1.0560 (1.0328,1.0798)
State 2 - State 4	0.04685 (0.02376, 0.09239)	1.0615 (1.0133,1.1119)
State 3 - State 3	-0.11946 (-0.17114,-0.08339)	
State 3 - State 4	0.11946 (0.08339, 0.17114)	1.0674 (1.0439,1.0914)

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is 0, which is not greater than replacement length

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is 0, which is not greater than replacement length

Misclassification probabilities

	Baseline
Obs State 2 State 2	0
Obs State 3 State 2	0.1041 (0.08898,0.1215)

Initial state occupancy probabilities

	Estimate	LCL	UCL
State 1	0.75681606	0.73449835	0.77726093
State 2	0.19930412	0.17966161	0.22057570
State 3	0.04387982	0.03430043	0.05598259
State 4	0.00000000	0.00000000	0.00000000

-2 * log-likelihood: 15059.61

[Note, to obtain old print format, use "printold.msm"]

ELECT summary

```
summary.elect(  
  model$LE, # life expectancy estimated by elect()  
  probs = c(.025, .5, .975), # numeric vector of probabilities for quantiles  
  digits=2, # number of decimals places in output  
  print = TRUE # print toggle  
)
```

ELECT summary

Covariates values in the multi-state model:

age
0

Covariates in the state-distribution model:

age

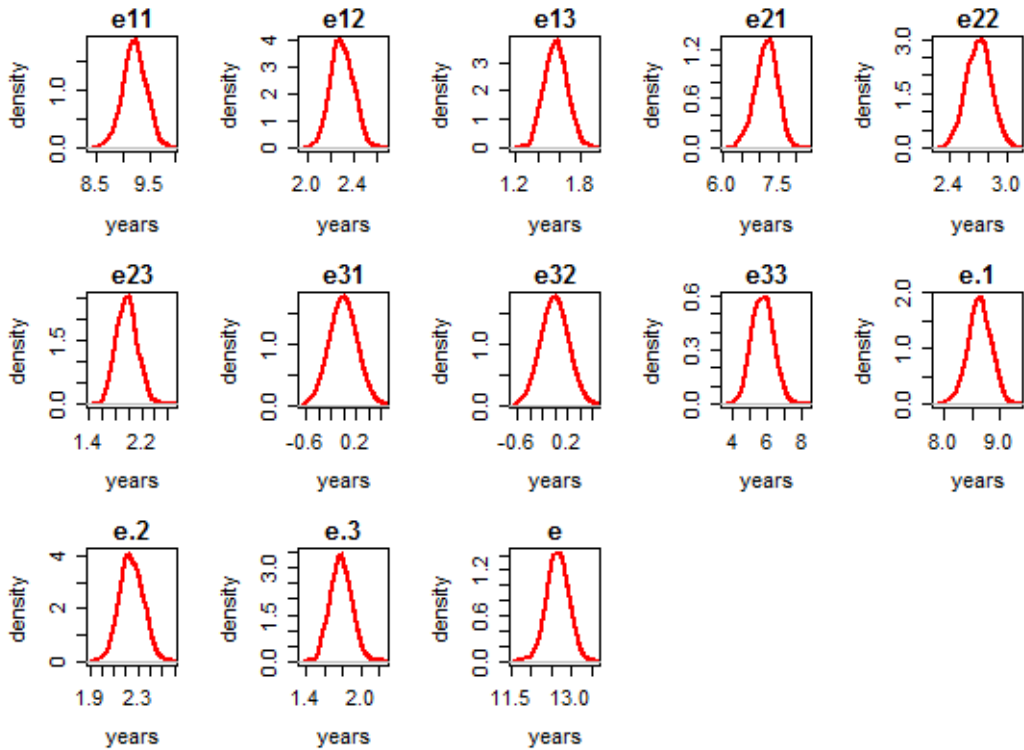
Life expectancies:Using simulation with 1000 replications

Point estimates, and mean, SEs, and quantiles from simulation:

	pnt	mn	se	0.025q	0.5q	0.975q
e11	9.26	9.23	0.22	8.78	9.24	9.64
e12	2.33	2.31	0.10	2.12	2.30	2.49
e13	1.58	1.57	0.10	1.38	1.57	1.77
e21	7.23	7.20	0.30	6.55	7.21	7.73
e22	2.74	2.71	0.13	2.45	2.71	2.95
e23	2.00	1.99	0.15	1.71	1.99	2.30
e31	0.00	0.00	0.00	0.00	0.00	0.00
e32	0.00	0.00	0.00	0.00	0.00	0.00
e33	5.75	5.76	0.62	4.62	5.75	6.98
e.1	8.67	8.64	0.21	8.21	8.65	9.04
e.2	2.27	2.25	0.09	2.07	2.24	2.44
e.3	1.79	1.79	0.11	1.57	1.79	2.01
e	12.73	12.68	0.27	12.16	12.68	13.18

plots

```
plot.elect(  
  model$LE, # life expectancy estimated by elect()  
  kernel = "gaussian", #character string for smoothing kernal ("gaussian",...)  
  col = "red", # color of the curve  
  lwd = 2, # line width of the curve  
  cex.lab = 1 # magnification to be used for axis-labels  
)
```



age at baseline

summary

```
model <- models[["age_bl"]]
msm_summary(model$msm)
```

```
-2loglik = 15026.84
Convergence code = 0
```

	p	se	Wald	ChiSq	Pr>ChiSq
qbase	-2.02	0.08	679.18	0.00	
qbase	-4.25	0.20	447.08	0.00	
qbase	-0.19	0.09	4.94	0.03	
qbase	-2.25	0.16	196.77	0.00	
qbase	-3.10	0.34	81.39	0.00	
qbase	-2.22	0.19	132.69	0.00	
qcov	0.04	0.01	6.96	0.01	
qcov	0.12	0.03	13.74	0.00	
qcov	-0.07	0.02	16.64	0.00	
qcov	0.08	0.02	12.07	0.00	
qcov	0.11	0.04	6.46	0.01	
qcov	0.09	0.02	20.14	0.00	
qcov	0.05	0.01	13.09	0.00	
qcov	-0.04	0.03	1.47	0.23	
qcov	0.06	0.02	10.64	0.00	
qcov	-0.03	0.02	1.49	0.22	
qcov	-0.07	0.05	2.30	0.13	

```

qcov  -0.04 0.02      2.65      0.10
p      -2.15 0.09     588.84     0.00
initp -1.33 0.06     433.67     0.00
initp -2.83 0.13     487.60     0.00

```

solution

```
print(model$msm, showEnv= F)
```

Call:

```
msm(formula = state ~ age, subject = id, data = ds, qmatrix = Q,      ematrix = E, covariates = covariate)
```

Maximum likelihood estimates

Baselines are with covariates set to 0

Transition intensities with hazard ratios for each covariate

	Baseline	age	age_bl
State 1 - State 1	-0.14696 (-0.168443,-0.12821)		
State 1 - State 2	0.13272 (0.114020, 0.15449)	1.0361 (1.0092,1.0638)	1.0519 (1.0234,1.081)
State 1 - State 4	0.01423 (0.009598, 0.02111)	1.1269 (1.0579,1.2004)	0.9605 (0.8999,1.025)
State 2 - State 1	0.82371 (0.694275, 0.97728)	0.9317 (0.9006,0.9639)	1.0628 (1.0246,1.102)
State 2 - State 2	-0.97424 (-1.130972,-0.83923)		
State 2 - State 3	0.10552 (0.077071, 0.14448)	1.0812 (1.0346,1.1299)	0.9720 (0.9287,1.017)
State 2 - State 4	0.04500 (0.022944, 0.08827)	1.1185 (1.0260,1.2194)	0.9332 (0.8534,1.020)
State 3 - State 3	-0.10897 (-0.158895,-0.07473)		
State 3 - State 4	0.10897 (0.074732, 0.15889)	1.0985 (1.0544,1.1445)	0.9652 (0.9249,1.007)

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is of replacement length

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is of replacement length

Misclassification probabilities

	Baseline
Obs State 2 State 2	0
Obs State 3 State 2	0.1041 (0.08896,0.1215)

Initial state occupancy probabilities

	Estimate	LCL	UCL
State 1	0.75610249	0.73399586	0.77561052
State 2	0.19930454	0.18017069	0.22017770
State 3	0.04459297	0.03486444	0.05662713
State 4	0.00000000	0.00000000	0.00000000

-2 * log-likelihood: 15026.84

[Note, to obtain old print format, use "printold.msm"]

ELECT summary

```
summary.elect(  
  model$LE, # life expectancy estimated by elect()  
  probs = c(.025, .5, .975), # numeric vector of probabilities for quantiles  
  digits=2, # number of decimals places in output  
  print = TRUE # print toggle  
)
```

ELECT summary

Covariates values in the multi-state model:

```
  age age_bl  
    0     0
```

Covariates in the state-distribution model:

```
  age
```

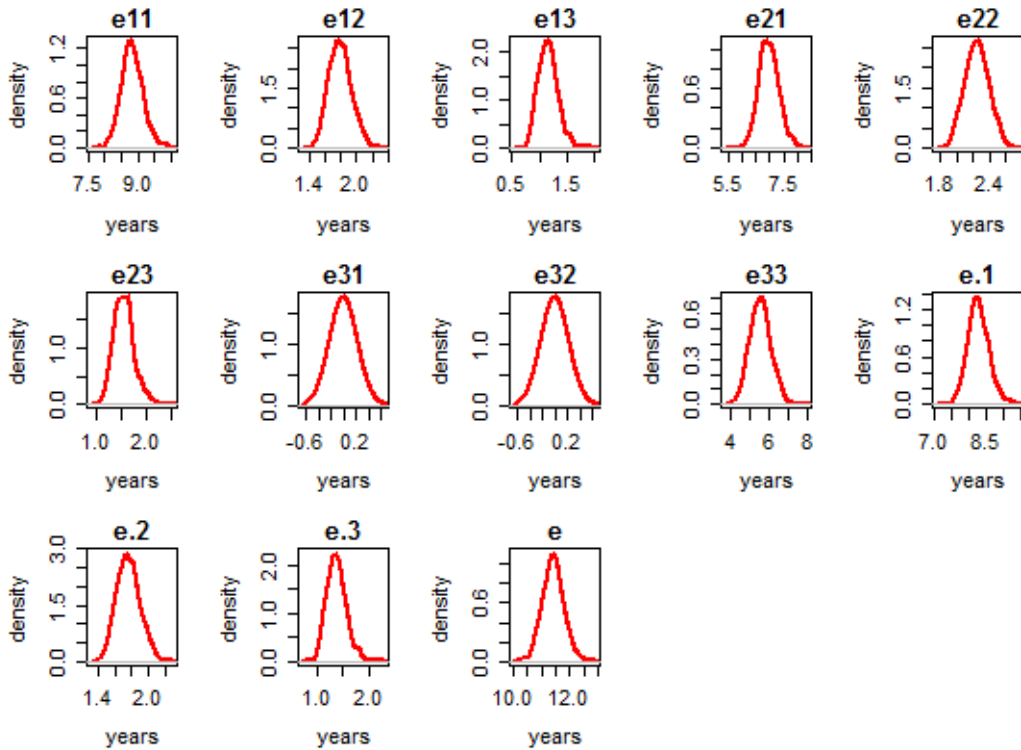
Life expectancies:Using simulation with 1000 replications

Point estimates, and mean, SEs, and quantiles from simulation:

	pnt	mn	se	0.025q	0.5q	0.975q
e11	8.87	8.83	0.32	8.23	8.81	9.50
e12	1.83	1.80	0.14	1.55	1.79	2.09
e13	1.17	1.16	0.18	0.87	1.15	1.56
e21	7.03	6.98	0.35	6.34	6.97	7.76
e22	2.29	2.25	0.14	1.99	2.25	2.53
e23	1.58	1.57	0.20	1.22	1.56	2.04
e31	0.00	0.00	0.00	0.00	0.00	0.00
e32	0.00	0.00	0.00	0.00	0.00	0.00
e33	5.58	5.58	0.55	4.54	5.58	6.69
e.1	8.32	8.28	0.31	7.71	8.26	8.92
e.2	1.80	1.77	0.13	1.53	1.76	2.04
e.3	1.39	1.39	0.18	1.08	1.38	1.79
e	11.51	11.43	0.37	10.75	11.43	12.18

plots

```
plot.elect(  
  model$LE, # life expectancy estimated by elect()  
  kernel = "gaussian", #character string for smoothing kernel ("gaussian",...)  
  col = "red", # color of the curve  
  lwd = 2, # line width of the curve  
  cex.lab = 1 # magnification to be used for axis-labels  
)
```

male

summary

```
model <- models[["male"]]
msm_summary(model$msm)
```

```
-2loglik = 14977.86
Convergence code = 0
```

	p	se	Wald ChiSq	Pr>ChiSq
qbase	-2.10	0.08	667.99	0.00
qbase	-4.38	0.21	437.79	0.00
qbase	-0.23	0.09	5.94	0.01
qbase	-2.23	0.17	179.55	0.00
qbase	-3.29	0.38	74.60	0.00
qbase	-2.47	0.21	132.89	0.00
qcov	0.04	0.01	6.79	0.01
qcov	0.13	0.03	15.41	0.00
qcov	-0.07	0.02	15.98	0.00
qcov	0.09	0.02	14.44	0.00
qcov	0.09	0.05	4.06	0.04
qcov	0.11	0.02	27.79	0.00
qcov	0.05	0.01	13.23	0.00
qcov	-0.05	0.03	2.12	0.15
qcov	0.06	0.02	10.04	0.00
qcov	-0.04	0.02	2.46	0.12
qcov	-0.05	0.05	0.87	0.35

```

qcov -0.05 0.02      5.80      0.02
qcov  0.33 0.09     13.14      0.00
qcov  0.45 0.23      3.77      0.05
qcov  0.12 0.11      1.30      0.25
qcov -0.16 0.16      1.08      0.30
qcov  0.64 0.28      5.13      0.02
qcov  0.43 0.14      9.23      0.00
p      -2.15 0.09    589.03      0.00
initp -1.33 0.06    434.12      0.00
initp -2.84 0.13    486.43      0.00

```

solution

```
print(model$msm, showEnv= F)
```

Call:

```
msm(formula = state ~ age, subject = id, data = ds, qmatrix = Q,      ematrix = E, covariates = covariates)
```

Maximum likelihood estimates

Baselines are with covariates set to 0

Transition intensities with hazard ratios for each covariate

	Baseline	age	age_bl	male
State 1 - State 1	-0.13442 (-0.155254,-0.11639)			
State 1 - State 2	0.12190 (0.103921, 0.14300)	1.0357 (1.0087,1.0635)	1.0523 (1.0238,1.0817)	1.3930 (1.3645,1.4215)
State 1 - State 4	0.01252 (0.008307, 0.01887)	1.1332 (1.0646,1.2062)	0.9528 (0.8927,1.0169)	1.5623 (1.5238,1.6008)
State 2 - State 1	0.79683 (0.663817, 0.95650)	0.9327 (0.9014,0.9651)	1.0612 (1.0229,1.1009)	1.1290 (1.0907,1.1673)
State 2 - State 2	-0.94185 (-1.104543,-0.80312)			
State 2 - State 3	0.10788 (0.077894, 0.14942)	1.0892 (1.0422,1.1383)	0.9642 (0.9212,1.0091)	0.8497 (0.8067,0.8927)
State 2 - State 4	0.03714 (0.017591, 0.07841)	1.0979 (1.0026,1.2024)	0.9558 (0.8692,1.0511)	1.8957 (1.8091,1.9823)
State 3 - State 3	-0.08480 (-0.128998,-0.05574)			
State 3 - State 4	0.08480 (0.055742, 0.12900)	1.1198 (1.0737,1.1679)	0.9489 (0.9093,0.9903)	1.5390 (1.4994,1.5786)

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is 0 of replacement length

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is 0 of replacement length

Misclassification probabilities

	Baseline
Obs State 2 State 2	0
Obs State 3 State 2	0.1042 (0.08905,0.1216)

Initial state occupancy probabilities

	Estimate	LCL	UCL
State 1	0.75644307	0.7336018	0.77663358
State 2	0.19914669	0.1799655	0.22019658
State 3	0.04441024	0.0350617	0.05654718
State 4	0.00000000	0.0000000	0.00000000

-2 * log-likelihood: 14977.86

[Note, to obtain old print format, use "printold.msm"]

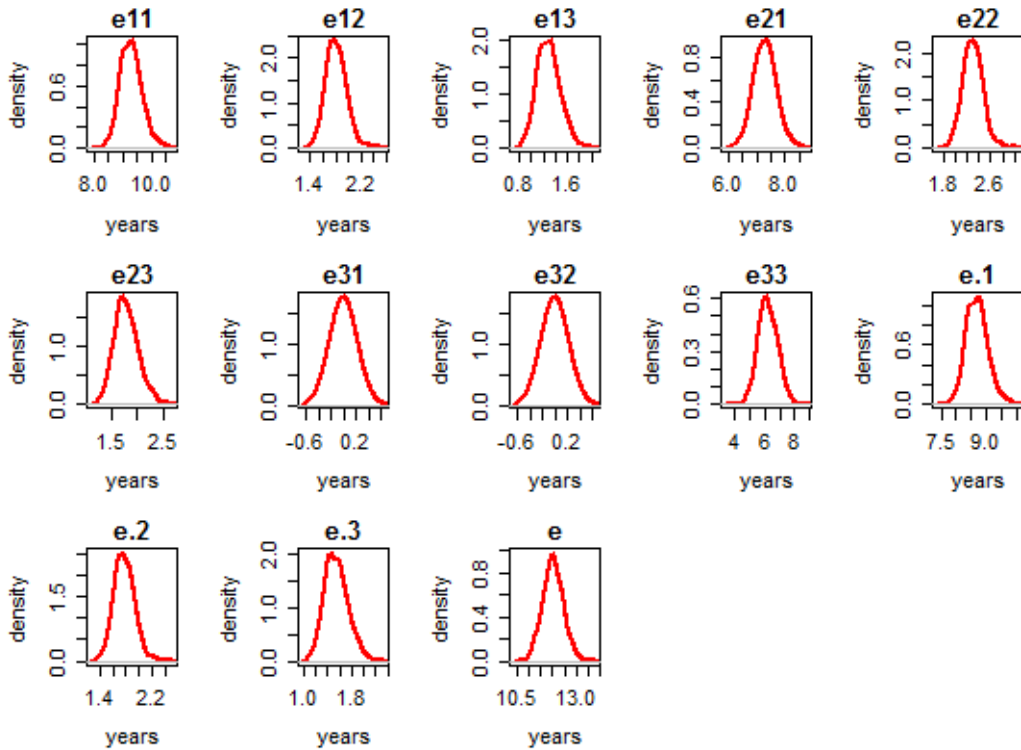
ELECT summary

```
summary.elect(  
  model$LE, # life expectancy estimated by elect()  
  probs = c(.025, .5, .975), # numeric vector of probabilities for quantiles  
  digits=2, # number of decimals places in output  
  print = TRUE # print toggle  
)
```

```
-----  
ELECT summary  
-----  
Covariates values in the multi-state model:  
  age age_bl  male  
    0     0     0  
Covariates in the state-distribution model:  
  age  
  
Life expectancies:Using simulation with 1000 replications  
  
Point estimates, and mean, SEs, and quantiles from simulation:  
      pnt    mn   se 0.025q 0.5q 0.975q  
e11  9.35  9.29 0.37   8.63  9.28 10.09  
e12  1.85  1.81 0.16   1.51  1.80  2.13  
e13  1.31  1.29 0.19   0.96  1.28  1.68  
e21  7.40  7.31 0.40   6.52  7.31  8.13  
e22  2.34  2.30 0.17   1.98  2.29  2.64  
e23  1.81  1.80 0.22   1.41  1.78  2.28  
e31  0.00  0.00 0.00   0.00  0.00  0.00  
e32  0.00  0.00 0.00   0.00  0.00  0.00  
e33  6.24  6.26 0.65   5.02  6.22  7.54  
e.1  8.76  8.70 0.35   8.08  8.69  9.47  
e.2  1.82  1.78 0.15   1.50  1.77  2.08  
e.3  1.57  1.55 0.19   1.22  1.54  1.96  
e   12.15 12.04 0.43  11.20 12.03 12.90  
-----
```

plots

```
plot.elect(  
  model$LE, # life expectancy estimated by elect()  
  kernel = "gaussian", #character string for smoothing kernel ("gaussian",...)  
  col = "red", # color of the curve  
  lwd = 2, # line width of the curve  
  cex.lab = 1 # magnification to be used for axis-labels  
)
```



education

summary

```
model <- models[["educat"]]
msm_summary(model$msm)
```

```
-2loglik = 14965.72
Convergence code = 0
```

	p	se	Wald ChiSq	Pr>ChiSq
qbase	-1.87	0.11	265.81	0.00
qbase	-4.32	0.32	186.83	0.00
qbase	-0.23	0.11	4.07	0.04
qbase	-2.15	0.19	125.35	0.00
qbase	-3.47	0.49	49.52	0.00
qbase	-2.38	0.22	112.26	0.00
qcov	0.04	0.01	6.79	0.01
qcov	0.12	0.03	14.88	0.00
qcov	-0.07	0.02	16.22	0.00
qcov	0.08	0.02	14.06	0.00
qcov	0.10	0.05	4.23	0.04
qcov	0.11	0.02	28.29	0.00
qcov	0.05	0.01	12.34	0.00
qcov	-0.05	0.03	1.87	0.17
qcov	0.06	0.02	10.14	0.00
qcov	-0.03	0.02	2.14	0.14
qcov	-0.05	0.05	1.08	0.30

```

qcov -0.05 0.02      5.71      0.02
qcov  0.33 0.09     12.92      0.00
qcov  0.49 0.23      4.41      0.04
qcov  0.12 0.11      1.30      0.25
qcov -0.14 0.16      0.81      0.37
qcov  0.57 0.30      3.63      0.06
qcov  0.44 0.14      9.66      0.00
qcov -0.25 0.09      7.31      0.01
qcov -0.09 0.27      0.11      0.74
qcov  0.02 0.11      0.03      0.87
qcov -0.10 0.14      0.56      0.45
qcov  0.25 0.44      0.33      0.56
qcov -0.15 0.12      1.39      0.24
p      -2.15 0.09    588.76      0.00
initp -1.33 0.06    433.80      0.00
initp -2.83 0.13    486.75      0.00

```

solution

```
print(model$msm, showEnv= F)
```

Call:

```
msm(formula = state ~ age, subject = id, data = ds, qmatrix = Q,      ematrix = E, covariates = covariates)
```

Maximum likelihood estimates

Baselines are with covariates set to 0

Transition intensities with hazard ratios for each covariate

	Baseline	age	age_bl	male
State 1 - State 1	-0.16778 (-0.205801,-0.13679)			
State 1 - State 2	0.15445 (0.123388, 0.19334)	1.0360 (1.0088,1.0640)	1.0510 (1.0222,1.0805)	1.3914 (1.3646,1.4182)
State 1 - State 4	0.01333 (0.007177, 0.02476)	1.1327 (1.0632,1.2067)	0.9546 (0.8931,1.0204)	1.6254 (1.5569,1.6939)
State 2 - State 1	0.79401 (0.634537, 0.99357)	0.9317 (0.9001,0.9643)	1.0617 (1.0233,1.1015)	1.1304 (1.0920,1.1688)
State 2 - State 2	-0.94182 (-1.146462,-0.77370)			
State 2 - State 3	0.11662 (0.080061, 0.16989)	1.0880 (1.0411,1.1371)	0.9662 (0.9228,1.0117)	0.8667 (0.8233,0.9101)
State 2 - State 4	0.03118 (0.011869, 0.08192)	1.1008 (1.0046,1.2063)	0.9492 (0.8603,1.0472)	1.7758 (1.6869,1.8647)
State 3 - State 3	-0.09296 (-0.144258,-0.05990)			
State 3 - State 4	0.09296 (0.059902, 0.14426)	1.1218 (1.0753,1.1703)	0.9490 (0.9091,0.9906)	1.5593 (1.5194,1.5992)
educat				
State 1 - State 1				
State 1 - State 2	0.7751 (0.6445,0.9323)			
State 1 - State 4	0.9152 (0.5427,1.5433)			
State 2 - State 1	1.0175 (0.8271,1.2518)			
State 2 - State 2				
State 2 - State 3	0.9007 (0.6848,1.1847)			
State 2 - State 4	1.2886 (0.5450,3.0466)			
State 3 - State 3				
State 3 - State 4	0.8635 (0.6765,1.1022)			

Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace is of replacement length

```
Warning in res[plabs == "pbase"][unique(states[plabs == "p"])] <- 1/(1 + : number of items to replace i
of replacement length
```

Misclassification probabilities

```

Baseline
Obs State 2 | State 2 0
Obs State 3 | State 2 0.1042 (0.08904,0.1216)
```

Initial state occupancy probabilities

```

Estimate      LCL      UCL
State 1 0.75627049 0.73396526 0.77637390
State 2 0.19927786 0.17984784 0.22039809
State 3 0.04445165 0.03500507 0.05651733
State 4 0.00000000 0.00000000 0.00000000
```

```
-2 * log-likelihood: 14965.72
```

```
[Note, to obtain old print format, use "printold.msm"]
```

ELECT summary

```
summary.select(
  model$LE, # life expectancy estimated by elect()
  probs = c(.025, .5, .975), # numeric vector of probabilities for quantiles
  digits=2, # number of decimals places in output
  print = TRUE # print toggle
)
```

----- ELECT summary -----

Covariates values in the multi-state model:

```
age age_bl male educat
0    0      0      0
```

Covariates in the state-distribution model:

```
age
```

Life expectancies:Using simulation with 1000 replications

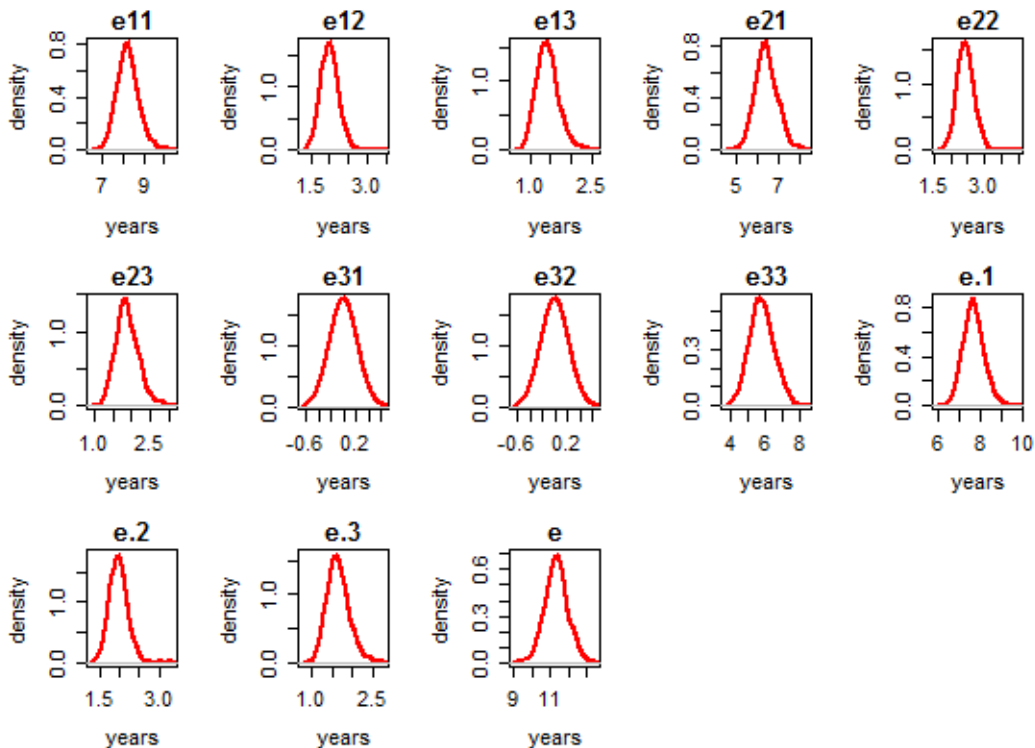
Point estimates, and mean, SEs, and quantiles from simulation:

	pnt	mn	se	0.025q	0.5q	0.975q
e11	8.33	8.24	0.51	7.31	8.22	9.33
e12	2.07	2.02	0.23	1.58	2.01	2.49
e13	1.45	1.43	0.26	1.00	1.41	1.98
e21	6.56	6.46	0.50	5.56	6.43	7.46
e22	2.51	2.45	0.24	1.99	2.44	2.94
e23	1.92	1.90	0.30	1.37	1.87	2.55
e31	0.00	0.00	0.00	0.00	0.00	0.00
e32	0.00	0.00	0.00	0.00	0.00	0.00
e33	5.84	5.84	0.69	4.57	5.83	7.25
e.1	7.80	7.72	0.48	6.85	7.69	8.75
e.2	2.03	1.97	0.22	1.56	1.97	2.43
e.3	1.68	1.66	0.26	1.20	1.64	2.23

```
e 11.51 11.36 0.60 10.22 11.36 12.55
```

plots

```
plot.elect(
  model$LE, # life expectancy estimated by elect()
  kernel = "gaussian", #character string for smoothing kernel ("gaussian",...)
  col = "red", # color of the curve
  lwd = 2, # line width of the curve
  cex.lab = 1 # magnification to be used for axis-labels
)
```



Session Info

```
sessionInfo()
```

```
R version 3.3.1 (2016-06-21)
```

```
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
Running under: Windows 10 x64 (build 14393)
```

```
locale:
```

```
[1] LC_COLLATE=English_United States.1252 LC_CTYPE=English_United States.1252
```

```
LC_MONETARY=English_U
```

```
[4] LC_NUMERIC=C
```

```
LC_TIME=English_United States.1252
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

other attached packages:

```
[1] msm_1.6.1    magrittr_1.5 nnet_7.3-12  knitr_1.14
```

loaded via a namespace (and not attached):

```
[1] Rcpp_0.12.6      formatR_1.4      nloptr_1.0.4      plyr_1.8.4        tools_3.3.1       dig
[7] lme4_1.1-12      evaluate_0.9      tibble_1.2        gtable_0.2.0      nlme_3.1-128      lat
[13] mgcv_1.8-14      Matrix_1.2-7.1    DBI_0.5           yaml_2.1.13       parallel_3.3.1    Spa
[19] mvtnorm_1.0-5    expm_0.999-0      dplyr_0.5.0       stringr_1.1.0     MatrixModels_0.4-1 gri
[25] R6_2.1.3         survival_2.39-5    rmarkdown_1.0     minqa_1.2.4       ggplot2_2.1.0     car
[31] scales_0.4.0     htmltools_0.3.5    splines_3.3.1     MASS_7.3-45       assertthat_0.1    pbk
[37] testit_0.5       colorspace_1.2-6  quantreg_5.26     stringi_1.1.1     lazyeval_0.2.0    mun
```