código imprimir

Mirna

2024-06-19

```
### Imputation of the age of the father ###########################################################

  ### Date: 8 June 2017

  ### Country: Sweden

  ### Description:
  ###

library(extrafont)
library(grid)
#font_import()
loadfonts()
loadfonts(device="win")
loadfonts(device="postscript")
loadfonts(device="pdf")

### Settings ######################################################################################

  rm(list=ls())

  years <- 1968:2014
  age.m <- 15:50
  age.f <- 15:50



### Read data #####################################################################################

  # Read birth data
  births <- read.table("N:/SGB/Male_Female_Fertility/Sweden/Tabel.csv",sep=",",header=T)
  # Simplify
  births <- aggregate(No_of_children~Age_of_Mother+Age_of_Father+Year,data=births,sum) #agrega (soma) "

  # Population counts
  pop <- read.table("U:/Data/HMD/sweden_population.txt",header=T)
  pop$Age <- as.numeric(as.character(pop$Age))
  pop <- pop[pop$Age%in%11:99,]



### Edit age ######################################################################################
```

```r
  births$Age_of_Mother <- as.character(births$Age_of_Mother)
  births$Age_of_Mother[births$Age_of_Mother=="-15"] <- "15"
  births$Age_of_Mother[births$Age_of_Mother=="50+"] <- "50"
  births$Age_of_Mother <- as.numeric(births$Age_of_Mother)

  births$Age_of_Father <- as.character(births$Age_of_Father)
  births$Age_of_Father[births$Age_of_Father=="-15"] <- "15"
  births$Age_of_Father[births$Age_of_Father=="50+"] <- "50"
  births$Age_of_Father <- as.numeric(births$Age_of_Father)



### Proportion missing over time ##################################################################

  # Objects for results
  prop.miss <- numeric(length(years))
  prop.missing.age <- matrix(data=NA,nrow=length(age.f),ncol=length(years))

  # Calculate
  for(i in years) {
    cat(".")
    prop.miss[i-(min(years)-1)] <- sum(births$No_of_children[births$Year==i&is.na(births$Age_of_Father)]

    for(j in age.f) {
      tmp <- births$No_of_children[births$Year==i&is.na(births$Age_of_Father)&births$Age_of_Mother==j]/
      if(length(tmp)==1&is.numeric(tmp)==T) prop.missing.age[j-min(age.f)+1,i-(min(years)-1)] <- tmp el
    }
  }

  # Correlation of proportion missing and age of mother
  tmpdat <- as.data.frame(cbind(rep(age.f,dim(prop.missing.age)[2]),c(prop.missing.age)))
  tmpdat$V1[is.nan(tmpdat$V1)] <- 0
  tmpdat$V2[is.nan(tmpdat$V2)] <- 0
  tmpdat <- na.omit(tmpdat)
  cor(tmpdat,method="spearman")


### Look at one year ##############################################################################

  ### Implements all imputation methods for one year
  ### Also brief assessment of age distributions

  # Set year to use
  whichyear <- 2014

  # Subset data
  birthsub <- subset(births,Year==whichyear&(Age_of_Father%in%age.m|is.na(Age_of_Father))&Age_of_Mother
  popsub1 <- subset(pop,Year==whichyear)
  popsub2 <- subset(pop,Year==whichyear+1)

  # Distribution of births for males and females (observed)
  tmp_f <- aggregate(No_of_children~Age_of_Mother,data=birthsub,sum)$No_of_children
  tmp_f2 <- aggregate(No_of_children~Age_of_Mother,data=birthsub[is.na(birthsub$Age_of_Father),],sum)$N
```

```
tmp_f_age <- aggregate(No_of_children~Age_of_Mother,data=birthsub,sum)$Age_of_Mother
tmp_f_age2 <- aggregate(No_of_children~Age_of_Mother,data=birthsub[is.na(birthsub$Age_of_Father),],sum

tmp_m <- aggregate(No_of_children~Age_of_Father,data=birthsub,function(x) sum(x,na.rm=T))$No_of_child
tmp_m_age <- aggregate(No_of_children~Age_of_Father,data=birthsub,function(x) sum(x,na.rm=T))$Age_of_F

tmp_m2 <- aggregate(No_of_children~Age_of_Father,data=birthsub[birthsub$Age_of_Mother==30,],function(
tmp_m_age2 <- aggregate(No_of_children~Age_of_Father,data=birthsub[birthsub$Age_of_Mother==30,],functi


# Mean age of fathers and mothers (obersved, dropping missings)
sum(tmp_f/sum(tmp_f)*tmp_f_age)
sum(tmp_m/sum(tmp_m)*tmp_m_age)

# Calculate TFR males (dropping missing values)
numerator.m <- numeric(length(age.m))
denominator.m <- numeric(length(age.m))
for(i in age.m) {
  numerator.m[i-(min(age.m)-1)] <- sum(birthsub$No_of_children[birthsub$Age_of_Father==i & !is.na(bir
  denominator.m[i-(min(age.m)-1)] <- (popsub2$Male[popsub2$Age==i]+popsub1$Male[popsub1$Age==i])/2
}
fertrate.m <- numerator.m/denominator.m

# TFR of females
numerator.f <- numeric(length(age.f))
denominator.f <- numeric(length(age.f))
for(i in age.f) {
  numerator.f[i-(min(age.f)-1)] <- sum(birthsub$No_of_children[birthsub$Age_of_Mother==i])
  denominator.f[i-(min(age.f)-1)] <- (popsub2$Female[popsub2$Age==i]+popsub1$Female[popsub1$Age==i])/
}
fertrate.f <- numerator.f/denominator.f

# TFR males: According to unconditional distribution
add.imp1 <- sum(birthsub$No_of_children[is.na(birthsub$Age_of_Father)])*(numerator.m/sum(numerator.m)
fertrate.m.imp1 <- (numerator.m+add.imp1)/denominator.m

# TFR males: According to fertility rates
add.imp2 <- sum(birthsub$No_of_children[is.na(birthsub$Age_of_Father)])*(fertrate.m/sum(fertrate.m))
fertrate.m.imp2 <- (numerator.m+add.imp2)/denominator.m

# TFR males: According to conditional distribution
add.imp3 <- numeric(length(age.m))

for(i in age.f) {
  btd <- sum(birthsub$No_of_children[birthsub$Age_of_Mother==i&is.na(birthsub$Age_of_Father)])
  if(btd==0) next # if no births to distribute
  if(btd==sum(birthsub$No_of_children[birthsub$Age_of_Mother==i])) next # if all births have NA
  agedistr <- numeric(length(age.m))
  tmp <- aggregate(No_of_children~Age_of_Father,data=birthsub[birthsub$Age_of_Mother==i,],sum)
  agedistr[match(tmp$Age_of_Father,age.m)] <- tmp$No_of_children
  agedistr <- agedistr/sum(agedistr)
  add.imp3 <- add.imp3+agedistr*btd
}
```

```r
  fertrate.m.imp3 <- (numerator.m+add.imp3)/denominator.m

  # Compare TFRs
  sum(fertrate.m)
  sum(fertrate.m.imp1)
  sum(fertrate.m.imp2)
  sum(fertrate.m.imp3)

  # Compare mean age
  sum(fertrate.m/sum(fertrate.m)*age.m)
  sum(fertrate.m.imp1/sum(fertrate.m.imp1)*age.m)
  sum(fertrate.m.imp2/sum(fertrate.m.imp2)*age.m)
  sum(fertrate.m.imp3/sum(fertrate.m.imp3)*age.m)

### Missingness pattern of data ####################################################################

  # Vector of proportion missing split by age of mother
  # I.e., sum(propsm) is the total proportion of missing values
  # This missingness pattern is used for the simulations
  match.ages <- match(birthsub$Age_of_Mother[is.na(birthsub$Age_of_Father)],age.f)
  propsm <- numeric(length(age.f))
  propsm[match.ages] <- birthsub$No_of_children[is.na(birthsub$Age_of_Father)]/sum(birthsub$No_of_child

  # Distribution of births by age of mother
  propsb <- numeric(length(age.f))
  tmp <- aggregate(No_of_children~Age_of_Mother,data=birthsub,sum)
  match.ages2 <- match(tmp$Age_of_Mother,age.f)
  propsb[match.ages2] <- tmp$No_of_children/sum(birthsub$No_of_children)



### Generate estimates of methods with varying proporiton missing ##################################

  # Proportion missings as goals
  ziele <- seq(0.01,0.5,by=0.005)
  missing.pattern <- matrix(data=NA,nrow=length(age.f),ncol=length(ziele))
  colnames(missing.pattern) <- paste(ziele)
  rownames(missing.pattern) <- paste(age.f)

  # Objects for results: TFRs
  itfr0 <- numeric(length(ziele))
  itfr1 <- numeric(length(ziele))
  itfr2 <- numeric(length(ziele))
  itfr3 <- numeric(length(ziele))

  # Mean age
  iage0 <- numeric(length(ziele))
  iage1 <- numeric(length(ziele))
  iage2 <- numeric(length(ziele))
  iage3 <- numeric(length(ziele))

  # Age specific fertility rates
  iasfr0 <- matrix(data=NA,ncol=length(ziele),nrow=length(age.m))
```

```
iasfr1 <- matrix(data=NA,ncol=length(ziele),nrow=length(age.m))
iasfr2 <- matrix(data=NA,ncol=length(ziele),nrow=length(age.m))
iasfr3 <- matrix(data=NA,ncol=length(ziele),nrow=length(age.m))

# Cycle over goals
for(j in ziele) {

  # Find missing vector
  tgoal <- j
  stopit <- F
  res1 <- propsm
  res2 <- propsm/propsb
  while(stopit==F) {

    tscale <- tgoal/sum(res1)
    usescale <- 0.9/res2
    usescale[usescale>tscale] <- tscale
    res1 <- usescale*res1
    res2 <- res1/propsb
    if(abs(sum(res1)-tgoal)<0.0001) stopit=T
  }

  missing.pattern[,which(ziele==j)] <- res2

  # Generate fake data
  birthfake <- subset(births,Year==whichyear&(Age_of_Father%in%age.m|is.na(Age_of_Father))&Age_of_Mot

  for(i in age.f) {
    total <- sum(birthfake$No_of_children[birthfake$Age_of_Mother==i])
    miss <- birthfake$No_of_children[birthfake$Age_of_Mother==i&is.na(birthfake$Age_of_Father)]
    valid <- birthfake$No_of_children[birthfake$Age_of_Mother==i&!is.na(birthfake$Age_of_Father)]
    pmiss <- sum(miss/total)
    pmgoal <- res2[i-(min(age.f)-1)]

    pvalid <- sum(valid/total)
    pvgoal <- 1-pmgoal

    pmscale <- pmgoal/pmiss
    pvscale <- pvgoal/pvalid

    birthfake$No_of_children[birthfake$Age_of_Mother==i&is.na(birthfake$Age_of_Father)] <- birthfake$
    birthfake$No_of_children[birthfake$Age_of_Mother==i&!is.na(birthfake$Age_of_Father)] <- birthfake
  }

  # Males (dropping missing values)
  fnumerator.m <- numeric(length(age.m))
  fdenominator.m <- numeric(length(age.m))
  for(i in age.m) {
    fnumerator.m[i-(min(age.m)-1)] <- sum(birthfake$No_of_children[birthfake$Age_of_Father==i & !is.na
    fdenominator.m[i-(min(age.m)-1)] <- (popsub2$Male[popsub2$Age==i]+popsub1$Male[popsub1$Age==i])/2
  }
  ifertrate.m.fake0 <- fnumerator.m/fdenominator.m
```

```r
    # According to unconditional distribution
    fadd.imp1 <- sum(birthfake$No_of_children[is.na(birthfake$Age_of_Father)])*(fnumerator.m/sum(fnumera
    ifertrate.m.fake1 <- (fnumerator.m+fadd.imp1)/fdenominator.m

    # According to fertility rates
    fadd.imp2 <- sum(birthfake$No_of_children[is.na(birthfake$Age_of_Father)])*(ifertrate.m.fake0/sum(if
    ifertrate.m.fake2 <- (fnumerator.m+fadd.imp2)/fdenominator.m

    # According to conditional distribution
    fadd.imp3 <- numeric(length(age.m))

    for(i in age.f) {
      btd <- sum(birthfake$No_of_children[birthfake$Age_of_Mother==i&is.na(birthfake$Age_of_Father)])
      if(btd==0) next # if no births to distribute
      if(btd==sum(birthsub$No_of_children[birthsub$Age_of_Mother==i])) next # if all births have NA
      agedistr <- numeric(length(age.m))
      tmp <- aggregate(No_of_children~Age_of_Father,data=birthfake[birthfake$Age_of_Mother==i,],sum)
      agedistr[match(tmp$Age_of_Father,age.m)] <- tmp$No_of_children
      agedistr <- agedistr/sum(agedistr)
      fadd.imp3 <- fadd.imp3+agedistr*btd
      if(any(is.nan(fadd.imp3))) stop("Was")
    }

    ifertrate.m.fake3 <- (fnumerator.m+fadd.imp3)/fdenominator.m

    # Generate results
    iasfr0[,which(ziele==j)] <- ifertrate.m.fake0
    iasfr1[,which(ziele==j)] <- ifertrate.m.fake1
    iasfr2[,which(ziele==j)] <- ifertrate.m.fake2
    iasfr3[,which(ziele==j)] <- ifertrate.m.fake3

    itfr0[which(ziele==j)] <- sum(ifertrate.m.fake0)
    itfr1[which(ziele==j)] <- sum(ifertrate.m.fake1)
    itfr2[which(ziele==j)] <- sum(ifertrate.m.fake2)
    itfr3[which(ziele==j)] <- sum(ifertrate.m.fake3)

    iage0[which(ziele==j)] <- sum(ifertrate.m.fake0/sum(ifertrate.m.fake0)*age.m)
    iage1[which(ziele==j)] <- sum(ifertrate.m.fake1/sum(ifertrate.m.fake1)*age.m)
    iage2[which(ziele==j)] <- sum(ifertrate.m.fake2/sum(ifertrate.m.fake2)*age.m)
    iage3[which(ziele==j)] <- sum(ifertrate.m.fake3/sum(ifertrate.m.fake3)*age.m)

  }



### Shifting distribution ##########################################################################

  ### This code generates "true" values for fertiltiy and compares it with the results from above

  # Goals for age shift
  amount_shift_list <- seq(-4,4,by=1)

  # Objects for results: 'True data'
```

```r
itfr_true_shift <- matrix(data=NA,ncol=length(ziele),nrow=length(amount_shift_list)) # True TFR
iage_true_shift <- matrix(data=NA,ncol=length(ziele),nrow=length(amount_shift_list)) # True mean age

# Objects for results:  Mean absolute percentage error by method
imape0 <- matrix(data=NA,ncol=length(ziele),nrow=length(amount_shift_list))
imape1 <- matrix(data=NA,ncol=length(ziele),nrow=length(amount_shift_list))
imape2 <- matrix(data=NA,ncol=length(ziele),nrow=length(amount_shift_list))
imape3 <- matrix(data=NA,ncol=length(ziele),nrow=length(amount_shift_list))

# Object for shifted age distributions (for mothers aged 30)
shifted_distr <- matrix(data=0,ncol=length(amount_shift_list),nrow=length(age.m))
rownames(shifted_distr) <- paste(age.m)
colnames(shifted_distr) <- amount_shift_list

for(amount_shift in amount_shift_list) {

  for(j in ziele) {

    # Find missing vector
    tgoal <- j
    stopit <- F
    res1 <- propsm
    res2 <- propsm/propsb
    while(stopit==F) {

      tscale <- tgoal/sum(res1)
      usescale <- 0.9/res2
      usescale[usescale>tscale] <- tscale
      res1 <- usescale*res1
      res2 <- res1/propsb
      if(abs(sum(res1)-tgoal)<0.0001) stopit=T
    }

    missing.pattern[,which(ziele==j)] <- res2

    # Generate fake data
    birthfake <- subset(births,Year==whichyear&(Age_of_Father%in%age.m|is.na(Age_of_Father))&Age_of_Mo

    for(i in age.f) {
      total <- sum(birthfake$No_of_children[birthfake$Age_of_Mother==i])
      miss <- birthfake$No_of_children[birthfake$Age_of_Mother==i&is.na(birthfake$Age_of_Father)]
      valid <- birthfake$No_of_children[birthfake$Age_of_Mother==i&!is.na(birthfake$Age_of_Father)]
      pmiss <- sum(miss/total)
      pmgoal <- res2[i-(min(age.f)-1)]

      pvalid <- sum(valid/total)
      pvgoal <- 1-pmgoal

      pmscale <- pmgoal/pmiss
      pvscale <- pvgoal/pvalid

      birthfake$No_of_children[birthfake$Age_of_Mother==i&is.na(birthfake$Age_of_Father)] <- birthfake
      birthfake$No_of_children[birthfake$Age_of_Mother==i&!is.na(birthfake$Age_of_Father)] <- birthfal
```

```
}

# True data sets Same age

  # Generate 'true' data, same age
  birthtrue_shift <- birthfake

  for(i in age.f) {
    tmp <- birthtrue_shift$No_of_children[birthtrue_shift$Age_of_Mother==i&is.na(birthtrue_shift$A
    if(length(tmp)==0) next
    distr <- numeric(length(age.m))
    obs.age <- birthtrue_shift$Age_of_Father[birthtrue_shift$Age_of_Mother==i&!is.na(birthtrue_shi
    if(length(obs.age)==0) next
    obs.counts <- birthtrue_shift$No_of_children[birthtrue_shift$Age_of_Mother==i&!is.na(birthtrue
    distr[match(obs.age,age.m)] <- obs.counts
    distr <- distr/sum(distr)
    shiftdistr <- numeric(length(age.m))
    for(z in age.m) {
      shiftdistr[max(min((z-14)+amount_shift,length(shiftdistr)),1)] <- shiftdistr[max(min((z-14)+
    }

    if(i==30) shifted_distr[,paste(amount_shift)] <- shiftdistr

    matching_ages <- match(obs.age,age.m)
    all_ages <- which(shiftdistr>0)
    notmatching_ages <- all_ages[!all_ages%in%matching_ages]

    birthtrue_shift$No_of_children[birthtrue_shift$Age_of_Mother==i&is.na(birthtrue_shift$Age_of_F
    birthtrue_shift$No_of_children[birthtrue_shift$Age_of_Mother==i&!is.na(birthtrue_shift$Age_of_

    if(length(notmatching_ages)>0) {
      nn <- length(notmatching_ages)
      tmp <- cbind(rep(i,nn),age.m[notmatching_ages],rep(whichyear,nn),tmp*shiftdistr[notmatching_
      colnames(tmp) <- names(birthtrue_shift)
      birthtrue_shift <- rbind(birthtrue_shift,tmp)
    }

  }

  # Calculate rate
  fnumerator.m <- numeric(length(age.m))
  fdenominator.m <- numeric(length(age.m))
  for(i in age.m) {
    fnumerator.m[i-(min(age.m)-1)] <- sum(birthtrue_shift$No_of_children[birthtrue_shift$Age_of_Fa
    fdenominator.m[i-(min(age.m)-1)] <- (popsub2$Male[popsub2$Age==i]+popsub1$Male[popsub1$Age==i]
  }
  ifertrate.m.true_shift <- fnumerator.m/fdenominator.m
  itfr_true_shift[which(amount_shift_list==amount_shift),which(ziele==j)] <- sum(ifertrate.m.true_
  iage_true_shift[which(amount_shift_list==amount_shift),which(ziele==j)] <- sum(ifertrate.m.true_
  imape0[which(amount_shift_list==amount_shift),which(ziele==j)] <- mean(abs((iasfr0[,which(ziele=
  imape1[which(amount_shift_list==amount_shift),which(ziele==j)] <- mean(abs((iasfr1[,which(ziele=
  imape2[which(amount_shift_list==amount_shift),which(ziele==j)] <- mean(abs((iasfr2[,which(ziele=
  imape3[which(amount_shift_list==amount_shift),which(ziele==j)] <- mean(abs((iasfr3[,which(ziele=
```

```
    }
  }
```