Hierarchical Poisson Model For Fertility Rate Prediction in Afghanistan

2025-05-18

```
install.packages("haven")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("coda")
install.packages("xtable")
install.packages("sf")
install.packages("latex2exp")
install.packages("AER")
install.packages("mvtnorm")
install.packages("extraDistr"j)
install.packages("VGAM")
```

Data preprocessing

Load, merge and select data

```
library(haven)
library(dplyr)
IMAGEOUT="./pics/BHP/"
# wm and hh are original data from UNICEF
# `arrange` to sort the data, won't change the result
wm <- read_sav("data_src/wm.sav")
wm <- wm %>% arrange(HH1, HH2, WM3)
hh <- read_sav("data_src/hh.sav")
hh <- hh %>% arrange(HH1, HH2)
# merge the data base on HH1 and HH2
wm_hh_merged <- merge(wm, hh, by = c("HH1", "HH2"))
## Check NA in CEB
# rows_with_na <- wm_hh_merged[is.na(wm_hh_merged$CM11), ]
# rows_with_na <- rows_with_na[is.na(rows_with_na$MA5), ]
variables <- c(
  'CM11',       # Children ever born
  'HH6.y',      # Area
  'HH7.y',      # Province
  'MA3',        # Husband has more wives
  'HH1',        # Cluster number
  'HH2',        # Household number
  'WM3',        # Woman's line number
  'WB6A',       # Highest level of school attended
  'MT2',        # Frequency of listening to the radio
  'MT3',        # Frequency of watching TV
  'MA2',        # Age of husband
  'WAGEM',      # Age at first marriage/union of woman
  'windex5.y',  # Wealth index quintile
  'HHAGE',      # Age of household head
  'helevel',    # Education of household head
  'HHSEX',      # Sex of household head
  'stratum.y',  # Stratum
  'wmweight',   # Woman's sample weight
  'WB4'         # Age of woman
)

# dataframe selected
df_sel <- wm_hh_merged[variables]

# rename the columns
colnames(df_sel) <- c(
  'CEB',        # Children ever born
  'area',       # Area
  'province',   # Province
  'other_wives',# Husband has more wives
  'HH1',        # Cluster number
  'HH2',        # Household number
  'WM3',        # Woman's line number
  'women_edu',  # Highest level of school attended
  'media_radio',# Frequency of listening to the radio
  'media_tv',   # Frequency of watching TV
  'husband_age',# Age of husband
  'women_agem', # Age at first marriage/union of woman
  'windex5',    # Wealth index quintile
  'HH_age',     # Age of household head
  'HH_edu',     # Education of household head
  'HH_sex',     # Sex of household head
  'stratum',    # Stratum
  'wgt',        # Woman's sample weight
  'women_age'   # Age of woman
)

# removes unmarried women
df_sel <- df_sel %>% filter(!is.na(CEB))
# removes 2 rows of missing data in the column `age at first marriage`
df_sel <- df_sel %>% filter(!is.na(women_agem))
```

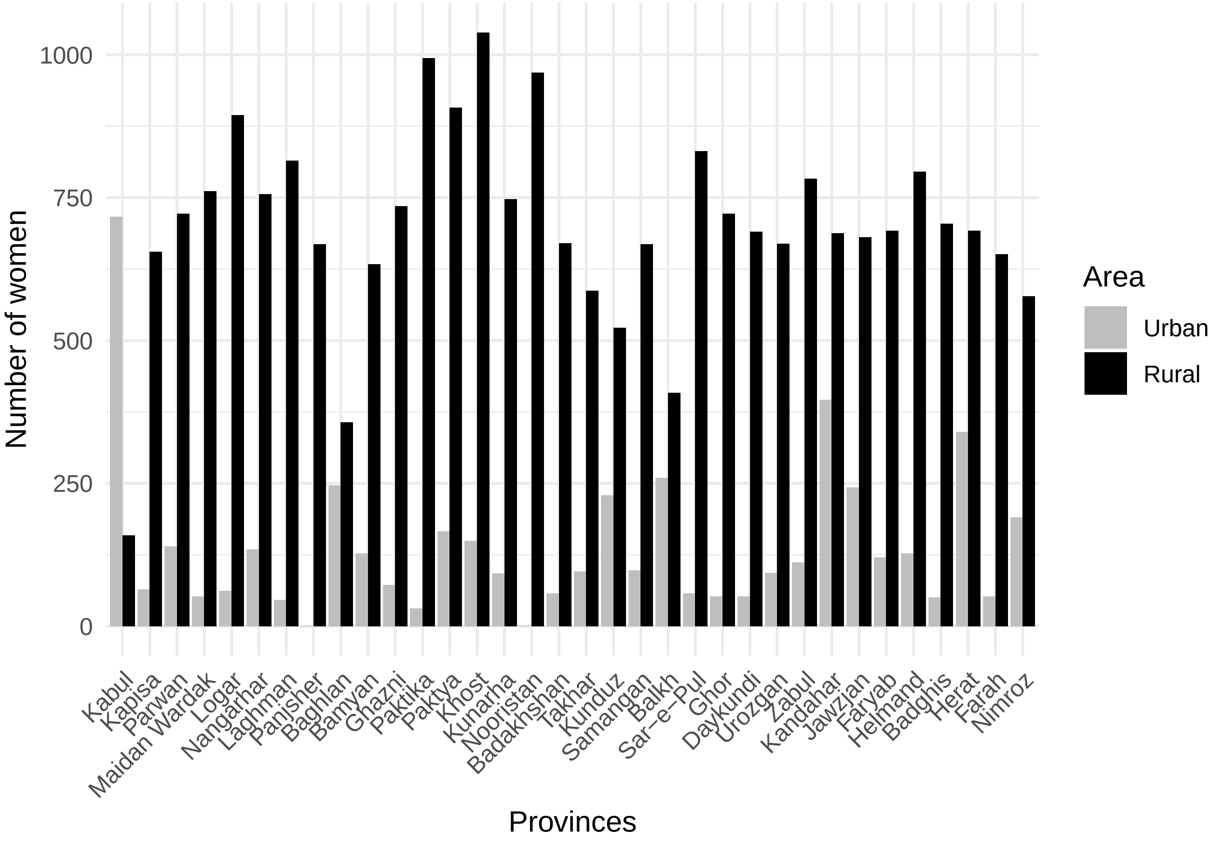
Data cleaning

```
# response variables
y <- df_sel$CEB

##############################################################
###################FIXED VARIABLES###########################
##############################################################

########################province##############################
province <- df_sel$province
```

```
   ##################################################
   #############Descriptive statistics###############
   ##################################################
table(province, useNA="ifany")
```

```
## province
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
##  876  719  862  813  956  890  861  668  603  760  807 1025 1073 1188  839  969  727  683  751  766
##   26   27   28   29   30   31   32   33   34
##  895 1084  924  812  922  754 1032  703  767
```

```
# Plot the number of records(women) we have in rural and urban area for each province
library(ggplot2)
province_vs_area <- with(df_sel,table(province, area))
province_vs_area_df <- as.data.frame(province_vs_area)
colnames(province_vs_area_df) <- c("Province", "Area", "Freq")
ggplot(province_vs_area_df, aes(x = Province, y = Freq, fill = Area))+
geom_bar(stat="identity", position="dodge")+
labs(
x = "Provinces",
y = "Number of women") +
scale_fill_manual(values = c("1" = "gray", "2" = "black"),
  labels = c("1" = "Urban", "2" = "Rural"))+
scale_x_discrete(
    labels = c(
    "1" = "Kabul",
    "2" = "Kapisa",
    "3" = "Parwan",
    "4" = "Maidan Wardak",
    "5" = "Logar",
    "6" = "Nangarhar",
    "7" = "Laghman",
    "8" = "Panjsher",
    "9" = "Baghlan",
    "10" = "Bamyan",
    "11" = "Ghazni",
    "12" = "Paktika",
    "13" = "Paktya",
    "14" = "Khost",
    "15" = "Kunarha",
    "16" = "Nooristan",
    "17" = "Badakhshan",
    "18" = "Takhar",
    "19" = "Kunduz",
    "20" = "Samangan",
    "21" = "Balkh",
    "22" = "Sar-e-Pul",
    "23" = "Ghor",
    "24" = "Daykundi",
    "25" = "Urozgan",
    "26" = "Zabul",
    "27" = "Kandahar",
    "28" = "Jawzjan",
    "29" = "Faryab",
    "30" = "Helmand",
    "31" = "Badghis",
    "32" = "Herat",
    "33" = "Farah",
    "34" = "Nimroz"
  )
)+
theme_minimal()+
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
dev.print(pdf,
    paste(IMAGEOUT,"NumOfRecords_ProvinceVsArea.pdf",sep=""),
    width=14,height=6)
```

```
## pdf
##   2
```

```
province <- factor(province)
# Using Herat as the reference level for province
province <- relevel(province, ref = 32)
province <- model.matrix(~province, data = province)
colnames(province) <- c(
  "P.HERAT.ref",
  "P.KABUL",
  "P.KAPISA",
  "P.PARWAN",
  "P.MAIDAN_WARDAK",
  "P.LOGAR",
  "P.NANGARHAR",
  "P.LAGHMAN",
  "P.PANJSHER",
  "P.BAGHLAN",
  "P.BAMYAN",
  "P.GHAZNI",
  "P.PAKTIKA",
  "P.PAKTYA",
  "P.KHOST",
  "P.KUNARHA",
  "P.NOORISTAN",
  "P.BADAKHSHAN",
  "P.TAKHAR",
  "P.KUNDUZ",
  "P.SAMANGAN",
  "P.BALKH",
  "P.SAR_E_PUL",
  "P.GHOR",
  "P.DAYKUNDI",
  "P.UROZGAN",
  "P.ZABUL",
  "P.KANDAHAR",
  "P.JAWZJAN",
  "P.FARYAB",
  "P.HELMAND",
  "P.BADGHIS",
  "P.FARAH",
  "P.NIMROZ"
)
########################province##############################

##########################area################################
area <- df_sel$area
area <- ifelse(area==2, "RURAL", "URBAN")
area <- factor(area)
area <- relevel(area, ref = "RURAL")
##########################area################################

###################husband has other wives####################
other_wives <- df_sel$other_wives
# remap, treat NA and 9(NO RESPONSE) the same as
# not having other wives
other_wives[is.na(other_wives)] <- 9
other_wives <- ifelse(other_wives > 1, 0, 1)
other_wives <- factor(other_wives)
# no need to relevel, reference level is 0 (no other wives)
# if you want to use 1 as reference level, set `ref=2`
# other_wives <- relevel(other_wives, ref = 1)
###################husband has other wives####################

###################household head sex#########################
# Remap, 0 for male; 1 for female
HH_sex <- case_when(
```

```r
    df_sel$HH_sex == 2 ~ 1,
    df_sel$HH_sex == 1 ~ 0
)
HH_sex <- factor(HH_sex)
##############################################################


##############################################################
####################RANDOM VARIABLES##########################
##############################################################

   ########################################################
   #################Quantitative Variables#################
   ########################################################

####################women's age###############################
women_age <- df_sel$women_age
####################women's age###############################

####################women's age squared#######################
women_age2 <- women_age^2
####################women's age squared#######################

###################household head's age#######################
HH_age <- df_sel$HH_age
###################household head's age#######################

###################household head's age squared###############
#HH_age2 <- HH_age^2
###################household head's age squared###############

###############women's age at marriage#########################
women_agem <- df_sel$women_agem
###############women's age at marriage#########################

   ########################################################
   #################Categorical  Variables#################
   ########################################################

####################women's edu level#########################
women_edu <- df_sel$women_edu
# `women_edu` comes from column `WB6A` which give the
# Highest level of edu level. So if a woman does not
# have any education, there will be a NA here.
# In fact, we have another column `welevel` on women's education,
# and the two columns give the same infomation.
# We can double check by comparing those two columns:
# View(cbind(wm$welevel,wm$WB6A))
# women that didnt go to school are going to have a 0 here
women_edu[is.na(women_edu)] <- 0
# remap class 5 (FORMAL ISLAMIC EDUCATION) to 0
women_edu <- ifelse(women_edu == 5, 0, women_edu)
# make it binary: 0 (for no education), 1 (has some education)
```

```r
women_edu <- ifelse(women_edu > 0, 1, 0)
women_edu <- factor(women_edu)
####################women's edu level#########################

###################household Head's edu level#################
HH_edu <- df_sel$HH_edu
# remap class 9 (DK/MISSING) to 0
HH_edu <- ifelse(HH_edu == 9, 0, HH_edu)
# make it binary: 0 (no education), 1 (has some education)
HH_edu <- ifelse(HH_edu > 0, 1, 0)
HH_edu <- factor(HH_edu)
###################household Head's edu level#################

###################wealth index (3 classes)###################
table(df_sel$windex5, useNA="ifany")
```

```
##
##    1    2    3    4    5
## 6096 6466 6388 5533 4080
```

```r
# combine the lower 2 levels as well as the higher 2 levels
windex3 <- case_when(
    df_sel$windex5 %in% c(1, 2) ~ 1,
    df_sel$windex5 == 3 ~ 2,
    df_sel$windex5 %in% c(4, 5) ~ 3
)
windex3 <- factor(windex3)
# we use the middle class as the reference level
windex3 <- relevel(windex3, ref=2)

windex3 <- model.matrix(~windex3, data = windex3)
colnames(windex3) <- c(
  "windex3.Middle.ref",
  "windex3.Poor",
  "windex3.Rich"
)
###################wealth index (3 classes)###################

########################media access##########################
media <- df_sel$media_tv + df_sel$media_radio
media <- ifelse(media >= 1, 1, 0)
media <- factor(media)
########################media access##########################
```

Hierarchical Poisson Model

Utility functions and data

```r
#######################utility functions######################
## mvnormal simulation
rmvnorm<-function(n,mu,Sigma)
{
  E<-matrix(rnorm(n*length(mu)),n,length(mu))
  t(  t(E%*%chol(Sigma)) +c(mu))
}

## Wishart simulation
rwish<-function(n,nu0,S0)
{
  sS0 <- chol(S0)
  S<-array( dim=c( dim(S0),n ) )
  for(i in 1:n)
  {
     Z <- matrix(rnorm(nu0 * dim(S0)[1]), nu0, dim(S0)[1]) %*% sS0
     S[,,i]<- t(Z)%*%Z
  }
  S[,,1:n]
}
## mvnorm log density
ldmvnorm<-function(X,mu,Sigma,iSigma=solve(Sigma),dSigma=det(Sigma))
{
  Y<-t( t(X)-mu)
  sum(diag(-.5*t(Y)%*%Y%*%iSigma))  -
  .5*(  prod(dim(X))*log(2*pi) +     dim(X)[1]*log(dSigma) )
}
## Calculate matrix multiplication: (1,Z) %*% Beta
matmul <- function(Z, Beta){
nZ <- nrow(Z)
as.matrix(cbind(rep(1,nZ),Z))%*%as.matrix(Beta)
}
#######################utility functions######################

#######################design matrix##########################
# Design matrix for all varables (random x10 | fixed x36)
X <- as.matrix(cbind(
    ### random predictors
  women_age,
  women_age2,
  HH_age,
#  HH_age2,
  women_agem,
  women_edu,
  HH_edu,
  windex3[,-1],
  media,
    ### fixed predictor
  province[,-1],# 33
  area,
  other_wives,
  HH_sex
))
# Design matrix for random variabe
# (the intercept will be add later)
Z <- as.matrix(cbind(
  women_age,
  women_age2,
  HH_age,
#  HH_age2,
  women_agem,
  women_edu,
  HH_edu,
  windex3[,-1],
  media
))

# Concat response variable y with design matrix of X
yX <- cbind(y,X)
########################design matrix##########################
```

Init

```r
########################initial values#########################
q<-dim(Z)[2] + 1# number of columns in Z (10) + 1 (intercept)
p<-dim(X)[2] + 1# number of columns in X (10+36) + 1 (intercept)
m<-length(unique(df_sel$stratum)) # number of stratum
# GAMMA is the (gamma_1, gamma_2,..., gamma_m)^T matrix
# with each gamma_j is a (1 x q) vector, sequentially stacked over rows
# So the dimension of GAMMA is (m x q)
# The initail value for gamma_j is set to zero because
# gamma_j ~ MVN(0,Sigma)
GAMMA <- matrix(0, nrow = m, ncol = q)

# eta0: inverse-Wishart prior for Sigma
eta0 <- q+2
# The initial mean of the MVN prior of theta
model <- glm(y~., data = data.frame(yX), family=poisson)
THETA <- mu0 <- model$coefficients

# The initial variance of the MVN prior for the variance of theta
# LAMBDA is the inverse of Fisher information matrix
LAMBDA <- vcov(model)

# another way to calculate LAMBDA, they are close but not the same
# ny <- dim(X)[1] # ny is the total number of rows
# X1 <- cbind(rep(1,ny), X) # X1 = (1|X)
# W <- diag(as.vector(X1%*%THETA))
# LAMBDA0 <- solve(t(X1)%*%W%*%X1)

# THETA1 stores m rows of ad hoc estimate of regression coefficients theta1
THETA1 <- NULL
# stratum_vector stores the labels of all 66 stratum
stratum_vector <- unique(df_sel$stratum)
# classical poisson regression on each strata
# traverse each stratum (total of 66 stratum)
for (stratum in stratum_vector){
stratum_data = yX[which(df_sel$stratum==stratum),]

captured <- tryCatch({
    model <- glm(y~., data = data.frame(stratum_data), family=poisson)
    THETA1<-rbind(THETA1, model$coeff[c(1:q)])
}, warning = function(w) {
    print(stratum)
    print(w)
})
#print(captured)
}

S0<-cov(THETA1)

# Prepare the list to store data in each stratum separately.
y.list <- list()
Z.list <- list()
X.list <- list()
wgt.list <- list()
N <- NULL
j <- 1
# traverse all 66 stratum
for (stratum in stratum_vector){
y.list[[j]] = y[which(df_sel$stratum==stratum)]
N[[j]]<-length(which(df_sel$stratum==stratum))
Z.list[[j]] = Z[which(df_sel$stratum==stratum),]
X.list[[j]] = X[which(df_sel$stratum==stratum),]
wgt.list[[j]] = df_sel[which(df_sel$stratum==stratum),c("wgt")]
j<-j+1
}

#summary(unlist(N))
#barplot(unlist(N))

# inverse Lambda_0, inverse S_0
iL0<-iSigma<-solve(S0)



## MCMC
# The accepted count of proposed gamma_j for each stratum (j from 1 to 66)
ACCEPT.count <- rep(0,m)
# The accepted count of proposed theta
ACCEPT.Theta.count <- 0
SIGMA.post<-NULL
GAMMA.post<- list()
THETA.post <- list()
S=220000
B=20000
thin=50
#######################initial values########################

######################debug##################################
lr_result <- NULL
delta_lr <- NULL
is_accepted <- NULL
theta_proposed <- NULL
```

```
#######################debug##################################
```

Simulation

```
#######################simulation#############################
set.seed(123)
start.time <- Sys.time()
for(s in 1:S)
{
    ##update Sigma
    iSigma<-rwish(1,eta0+m, solve( S0+t(GAMMA)%*%(GAMMA)) )

    ##update theta
    # theta proposed
    K_theta <- 0.05
    theta.p<-t(rmvnorm(1,THETA,K_theta*LAMBDA))
    lr<-0
    for(j in 1:m)
    {
        lr <- lr+sum(
            wgt.list[[j]] * dpois(y.list[[j]],
                lambda=exp(matmul(Z.list[[j]],GAMMA[j, ]) + matmul(X.list[[j]],theta.p)),log=TRUE),
            -wgt.list[[j]] * dpois(y.list[[j]],
                lambda=exp(matmul(Z.list[[j]],GAMMA[j, ]) + matmul(X.list[[j]],THETA)),log=TRUE)
        )
    }
  dlr <- ldmvnorm(t(theta.p),mu0,LAMBDA) - ldmvnorm(t(THETA),mu0,LAMBDA)
    lr <- lr + dlr
  ################debug###############
  delta_lr <- c(delta_lr, dlr)
  lr_result <- c(lr_result, lr)
  theta_proposed <- rbind(theta_proposed,t(theta.p))
  #cat("delta_lr is: ")
    #cat(delta_lr)
  #cat(" lr is: ")
    #print(lr)
  ################debug###############
    if( log(runif(1))<lr ) {
  ################debug###############
    is_accepted <- c(is_accepted, TRUE)
  ################debug###############
        ACCEPT.Theta.count<- ACCEPT.Theta.count+1
        THETA<-theta.p
    }else{
  ################debug###############
    is_accepted <- c(is_accepted, FALSE)
  ################debug###############
  }

    ##update gamma
    Sigma<-solve(iSigma) ; dSigma<-det(Sigma)
    K_gamma = rep(0.2,m)
    K_gamma[1] <- 0.05
    K_gamma[6] <- 0.1
    K_gamma[16] <- 0.1
    K_gamma[29] <- 0.1
    K_gamma[31] <- 0.1
    K_gamma[37] <- 0.1
    K_gamma[51] <- 0.1
    K_gamma[60] <- 0.1
    K_gamma[55] <- 0.1
    K_gamma[57] <- 0.05
    K_gamma[60] <- 0.1
    K_gamma[61] <- 0.1
    # PD <- outer((sqrt(K_gamma)),(sqrt(K_gamma)) ,"*")
    # We don't need a 66x66 matrix PD to be the scaling factor:
    # For one thing the PD %*% Sigma is non-conformable,
    # and also it's enough to use a scalar as the scaling factor
    # https://www.maths.lancs.ac.uk/~sherlocc/Publications/rwm.final.pdf

    for(j in 1:m)
    {
        # beta proposed (beta^(*)) from multivariate normal (beta_j^(s), V_J^(s)
      gamma.p<-t(rmvnorm(1,GAMMA[j,],K_gamma[j]*Sigma))
      lr<-sum(
            wgt.list[[j]] * dpois(y.list[[j]],
                lambda=exp(matmul(Z.list[[j]],gamma.p) + matmul(X.list[[j]],THETA)),log=TRUE),
            -wgt.list[[j]] * dpois(y.list[[j]],
                lambda=exp(matmul(Z.list[[j]],GAMMA[j, ]) + matmul(X.list[[j]],THETA)),log=TRUE),
            ldmvnorm( t(gamma.p),rep(0,q),Sigma,iSigma=iSigma,dSigma=dSigma ),
            -ldmvnorm( t(GAMMA[j,]),rep(0,q),Sigma,iSigma=iSigma,dSigma=dSigma )
        )

      if( log(runif(1))<lr ) {
            #print(sprintf("Beta %d Accepted", j ))
            ACCEPT.count[j] <- ACCEPT.count[j]+1
            GAMMA[j,]<-gamma.p
        }
    }

    ##store some output
    if((s %% thin == 0) & (s > B)) # saving every 10th value
    {
        cat("Current iteration index = ", s, "\n")
        # Print current acceptance rate
        cat("Current acceptance rate of Theta = ", round((ACCEPT.Theta.count/s)*100, digits = 2),"\n")
        # save thinned coefficients
        THETA.post<-rbind(THETA.post,t(THETA))
        SIGMA.post<-rbind(SIGMA.post,c(Sigma))
        GAMMA.post[[length(GAMMA.post)+1]]<-GAMMA
    }
}
end.time <- Sys.time()
#######################simulation##########################
```

Save and load results

```
RESULTPATH = "./data_output/BHP/"
save(THETA.post, file=paste(RESULTPATH,"THETA.post.S=",S,".RData",sep=""))
save(SIGMA.post, file=paste(RESULTPATH,"SIGMA.post.S=",S,".RData",sep=""))
save(ACCEPT.count, file=paste(RESULTPATH,"ACCEPT.count.S=",S,".RData",sep=""))
save(ACCEPT.Theta.count, file=paste(RESULTPATH,"ACCEPT.Theta.count.S=",S,".RData",sep=""))
save(GAMMA, file=paste(RESULTPATH,"GAMMA.S=",S,".RData",sep=""))
save(GAMMA.post, file=paste(RESULTPATH,"GAMMA.post.S=",S,".RData",sep=""))
```

```
RESULTPATH = "./data_output/BHP/"
load(paste(RESULTPATH,"THETA.post.S=",S,".RData",sep=""))
load(paste(RESULTPATH,"SIGMA.post.S=",S,".RData",sep=""))
load(paste(RESULTPATH,"ACCEPT.count.S=",S,".RData",sep=""))
load(paste(RESULTPATH,"ACCEPT.Theta.count.S=",S,".RData",sep=""))
load(paste(RESULTPATH,"GAMMA.S=",S,".RData",sep=""))
load(paste(RESULTPATH,"GAMMA.post.S=",S,".RData",sep=""))
```

Plot results

Assess the convergence of the Markov chain

```
## stationarity plot - boxplot
stationarity.plot<-function(x,...){
S<-length(x)
scan<-1:S
ng<-min( round(S/100),10)
group<-S*ceiling( ng*scan/S) /ng
boxplot(x~group,...)
}
```

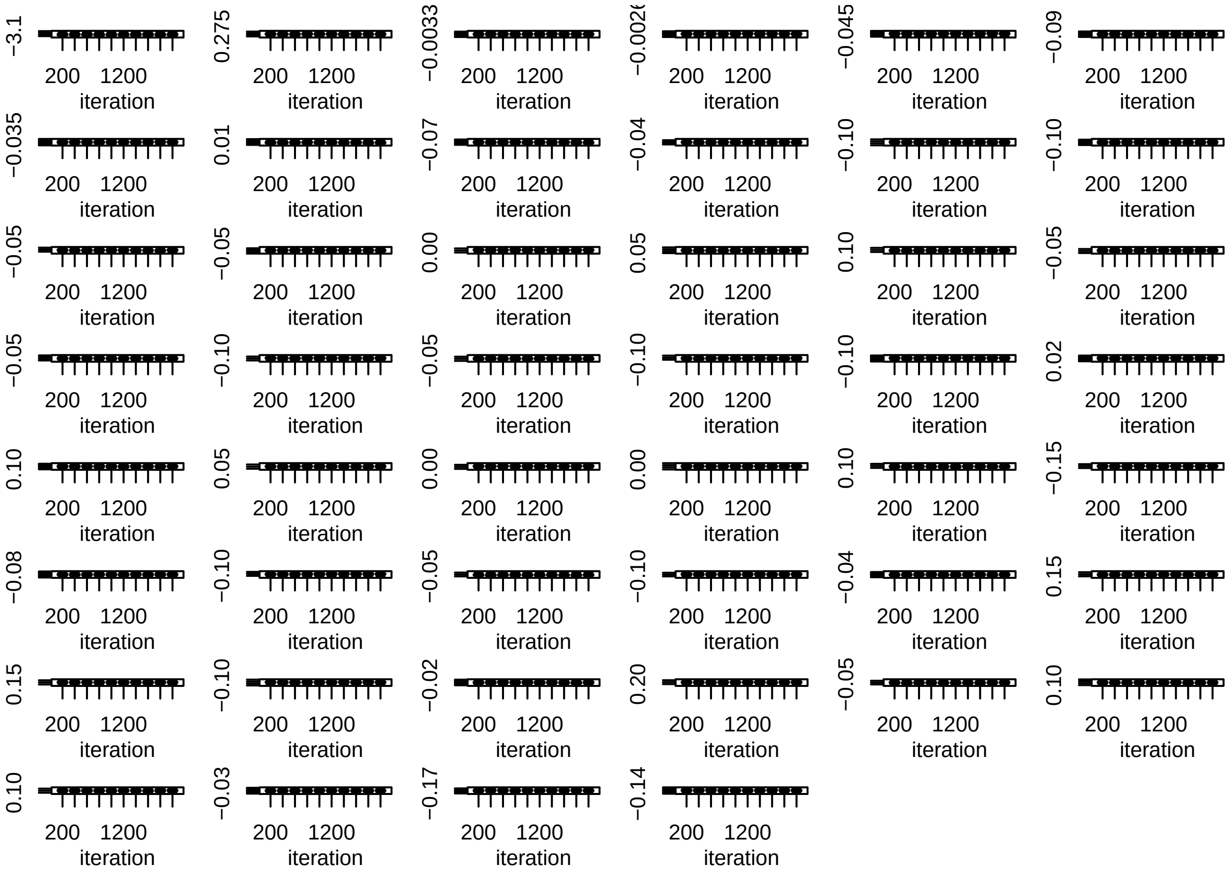
```
# convert the result (from list) into a 47x(S-B) matrix
# thin again, set thin2 = 1 means we don't thin again
thin2 = 2
THETA.post.mat <- NULL
for (i in c(1:dim(THETA.post)[1])){
    if(i %% thin2 == 0){
        # Thin again
        THETA.post.mat <- cbind(THETA.post.mat, as.matrix(unlist(t(THETA.post)[,i])) )
    }
}
GAMMA.post.mat <- list()
j <- 1
for (stratum in stratum_vector){
    gamma_j <- NULL
    for (s in c(1:length(GAMMA.post))){
        if(i %% thin2 == 0){
        # Thin again
            gamma_j <- rbind(gamma_j, GAMMA.post[[s]][j, ])
        }
    }
    GAMMA.post.mat[[j]] <- gamma_j
j<-j+1
}
# GAMMA.post.mat Usage: matmul(Z, apply(GAMMA.post.mat[[9]],2,mean))
SIGMA.post.mat <- NULL
for (i in c(1:dim(SIGMA.post)[1])){
    if(i %% thin2 == 0){
        # Thin again
        SIGMA.post.mat <- cbind(SIGMA.post.mat, SIGMA.post[i,] )
    }
}

ACCEPT.count/S
```

```
##  [1] 0.3109318 0.2797636 0.4850591 0.3093455 0.3782818 0.2550318 0.4847000 0.2448045 0.4497636 0.348
## [13] 0.3512500 0.2325545 0.4746182 0.3591045 0.4960909 0.2993682 0.4541591 0.4767591 0.2803227 0.432
## [25] 0.4608182 0.3002091 0.4153227 0.4333182 0.3483909 0.3878136 0.3567409 0.3355955 0.2320500 0.458
## [37] 0.3249727 0.4672227 0.2754364 0.4409500 0.4588364 0.2779318 0.4650136 0.2509545 0.2730364 0.470
## [49] 0.3154773 0.2625955 0.3457682 0.3923591 0.2919182 0.3952455 0.3177364 0.3490864 0.3654682 0.465
## [61] 0.2336636 0.4860273 0.4731318 0.2881773 0.4399955 0.3749136
```

```
#stationarity.plot(THETA.post[, 1],xlab="iteration",ylab=expression(theta[11]))
par(mfrow = c(8, 6), mar=c(3,2,1,1),mgp=c(2,1,0))
for (i in 1:p) {
stationarity.plot(THETA.post.mat[i, ],xlab="iteration",ylab=expression(theta))
}
dev.print(pdf,
    paste(IMAGEOUT,"THETA.post.boxplot.pdf",sep=""),
    width=16,height=12)
```

```
## pdf
##   2
```

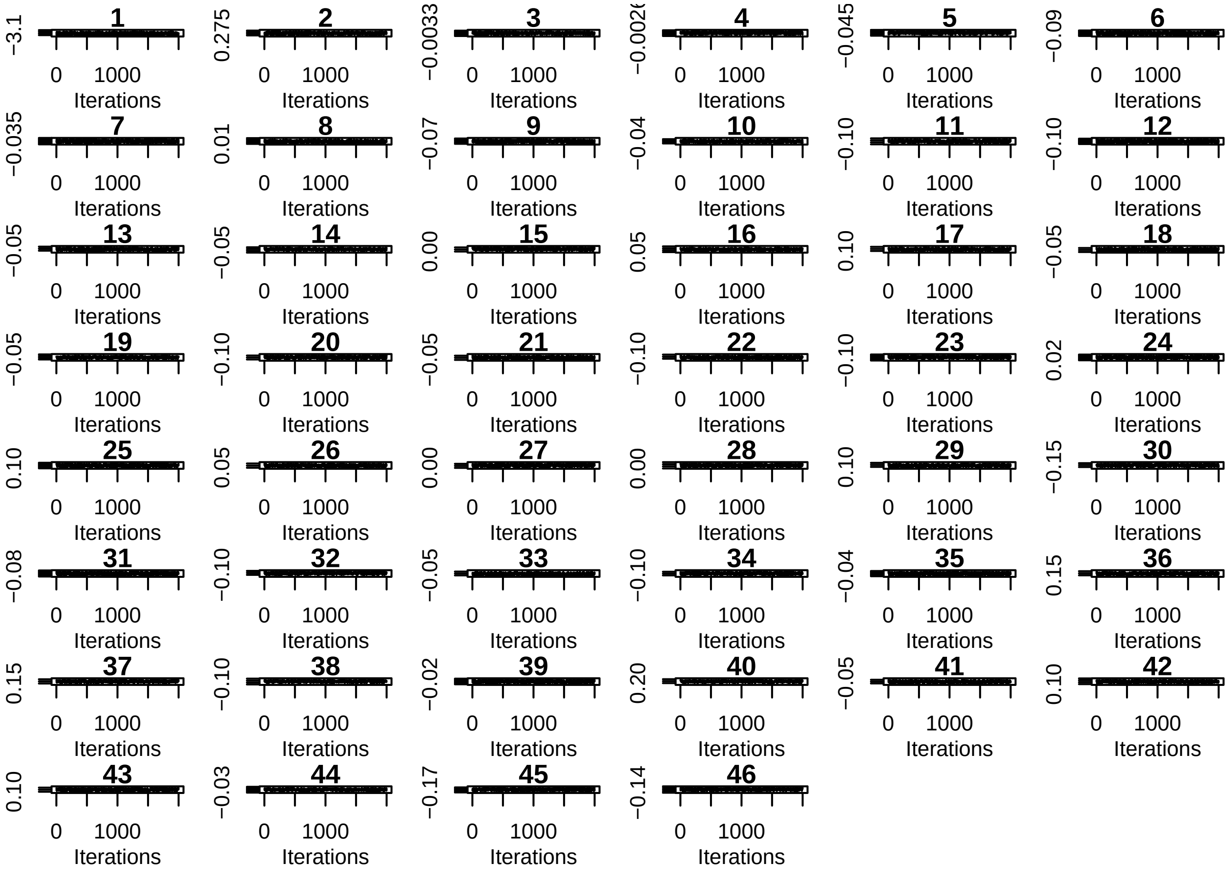
```
# Traceplot
par(mfrow = c(8, 6), mar=c(3,2,1,1),mgp=c(2,1,0))
```



```
for (i in 1:p) {
plot(seq(1,dim(THETA.post.mat)[2],by=1), THETA.post.mat[i,], type = "l", xlab = "Iterations", ylab = exp
}
dev.print(pdf,
    paste(IMAGEOUT,"THETA.post.pdf",sep=""),
    width=16,height=12)
```

```
## pdf
##   2
```

```
# ACF plot
library(coda)
par(mfrow = c(8, 6), mar=c(3,2,1,1),mgp=c(2,1,0))
```



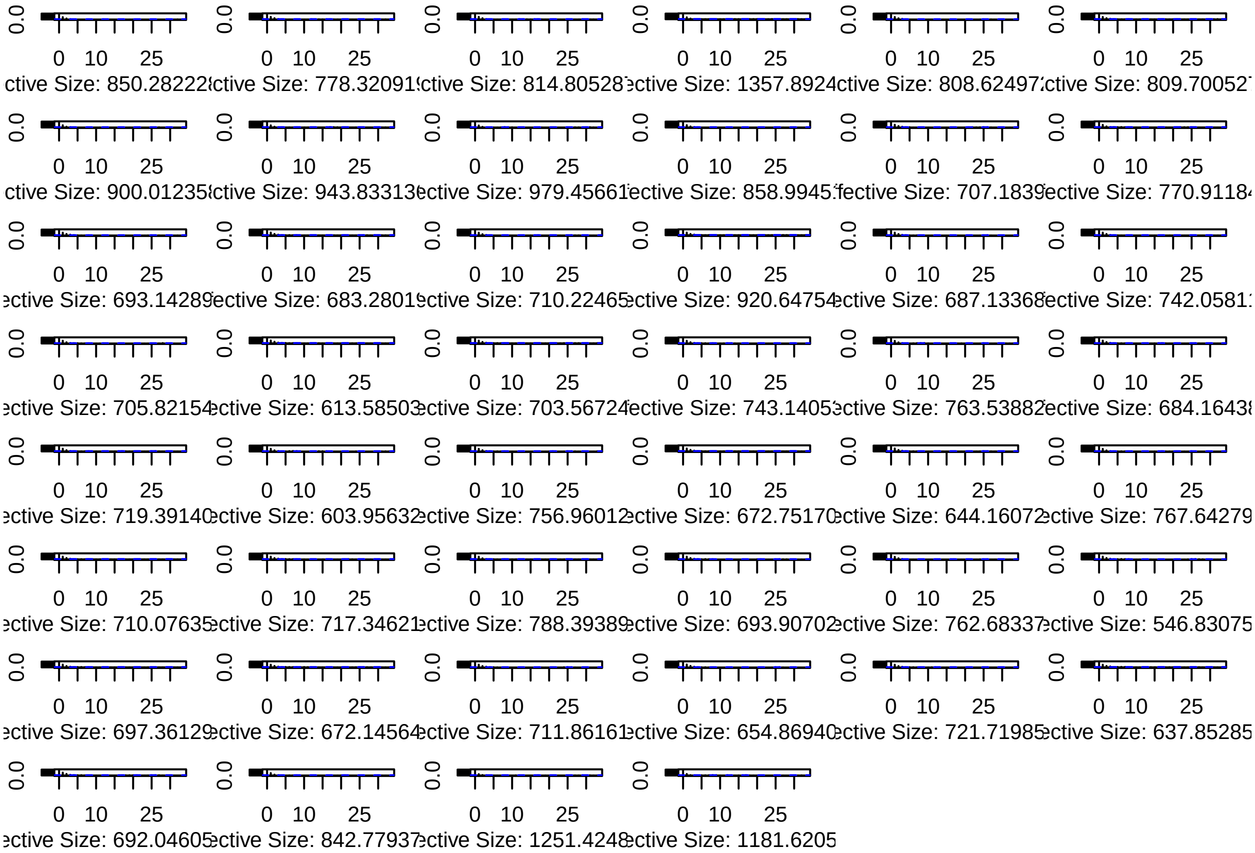
```
for (i in 1:p){
esize <- effectiveSize(THETA.post.mat[i,])
acf_plot <- acf(THETA.post.mat[i,], plot = FALSE)
t <- paste(i,",Effective Size: ", esize, sep="")
plot(acf_plot, xlab=t, ylab="", main=t)
}
dev.print(pdf,
    paste(IMAGEOUT,"ACF.THETA.post.pdf",sep=""),
    width=16,height=12)
```

```
## pdf
##   2
```

```
apply(SIGMA.post, 2, effectiveSize)
```

```
##    [1] 1558.1344 1591.7707 1579.6102 1373.9715 1277.5433 1341.1605 1430.1154 1437.6407 1101.1437 1381
##   [13] 1750.6108 1220.0073 1470.7468 1308.1301 1246.5758 1609.1978  988.1471 1394.4397 1579.6102 1750
##   [25] 1504.9405 1381.6452 1353.5069 1501.6989  934.5862 1382.7096 1373.9715 1220.0073 1227.0044 1433
##   [37] 1529.8688 1562.9577 1337.3888 1247.5702 1277.5433 1470.7468 1504.9405 1514.4510 2633.9032 2031
##   [49]  820.2251 1102.8041 1341.1605 1308.1301 1381.6452 1267.0094 2031.2272 1664.2404 1675.0933 1495
##   [61] 1430.1154 1246.5758 1353.5069 1529.8688 1365.4155 1675.0933 1565.9235 1554.3854 1396.0651 1481
##   [73] 1501.6989 1562.9577 1797.1388 1495.6372 1554.3854 1862.0843 1426.4496 1412.4410 1101.1437  988
##   [85]  820.2251 1152.5892 1396.0651 1426.4496 1374.3209 1238.0596 1381.4869 1394.4397 1382.7096 1247
##   [97] 1481.3000 1412.4410 1238.0596 1465.0921
```

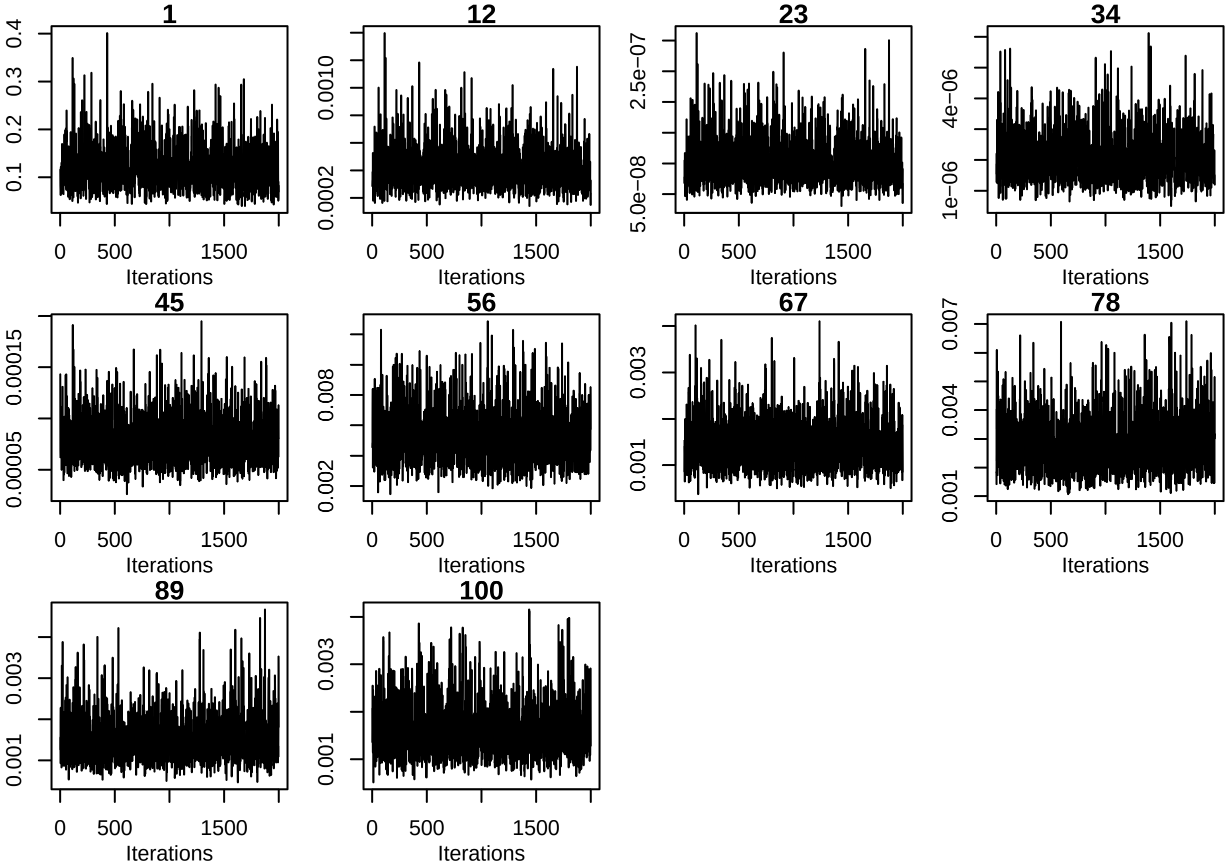
```
# index of sigma
ind_sigma <- c(1:q)
for(i in 1:q){
  ind_sigma[i] <- i+q*(i-1)
}

par(mfrow=c(3,4))
```



```
for(i in ind_sigma){
plot(seq(1,dim(SIGMA.post.mat)[2],by=1), SIGMA.post.mat[i,], type = "l", xlab = "Iterations", ylab = exp
}
dev.print(pdf,
    paste(IMAGEOUT,"SIGMA.post.pdf",sep=""),
    width=16,height=12)
```

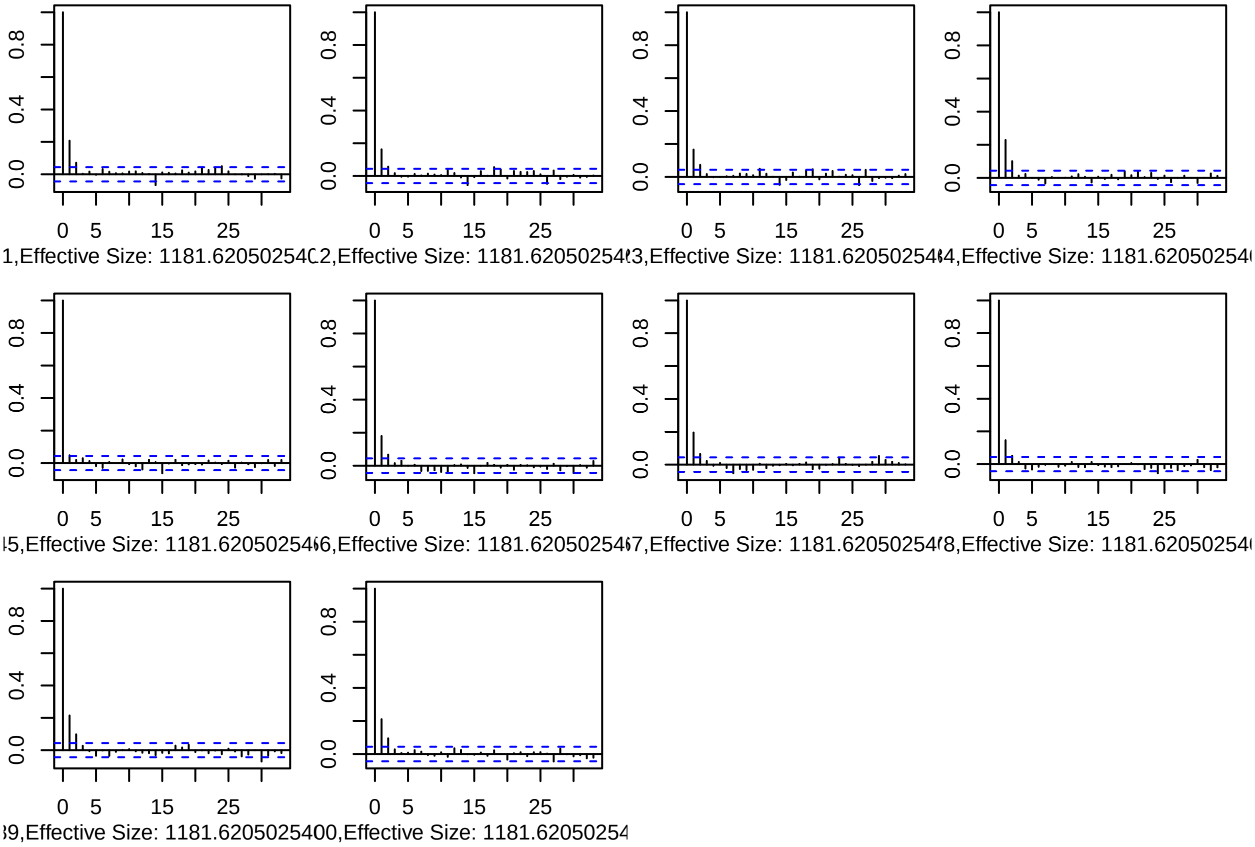
```
## pdf
##   2
```

```
par(mfrow=c(3,4))
```



```
for(i in ind_sigma){
acf_plot <- acf(SIGMA.post.mat[i,], plot=FALSE)
    t=paste(i,",Effective Size: ", esize, sep="")
plot(acf_plot, xlab=t, ylab="", main=expression(paste("acf(",sigma^2,")",sep="")))
}
dev.print(pdf,
    paste(IMAGEOUT,"ACF.SIGMA.post.pdf",sep=""),
    width=16,height=12)
```

```
## pdf
##   2
```



Posterior estimates with 95% credible intervals (CI) of the incidence rate ratio (IRR) for the explanatory variables

```
library(xtable)
CI_THETA <- round(t(apply(exp(THETA.post.mat), 1, quantile ,probs=c(0.50, 0.025, 0.975))), digits = 5)
CI_THETA
```

```
##                     50%    2.5%   97.5%
## (Intercept)     0.04961 0.04472 0.05510
## women_age       1.32568 1.31886 1.33287
## women_age2      0.99684 0.99676 0.99691
## HH_age          0.99807 0.99770 0.99841
## women_agem      0.95800 0.95681 0.95909
## women_edu       0.93764 0.92225 0.95423
## HH_edu          0.98247 0.97256 0.99308
## windex3.Poor    1.02722 1.01448 1.03972
## windex3.Rich    0.95007 0.93727 0.96298
## media           0.97518 0.96496 0.98617
## P.KABUL         0.97743 0.93620 1.02020
## P.KAPISA        0.96199 0.92087 1.00786
## P.PARWAN        0.98474 0.94548 1.03009
## P.MAIDAN_WARDAK 1.04346 0.99991 1.09099
## P.LOGAR         1.06277 1.01910 1.10885
## P.NANGARHAR     1.12731 1.08343 1.17294
## P.LAGHMAN       1.14648 1.10008 1.19693
## P.PANJSHER      1.01206 0.96681 1.05917
## P.BAGHLAN       1.01186 0.96628 1.05976
## P.BAMYAN        0.95855 0.91566 1.00256
## P.GHAZNI        1.00418 0.96342 1.05000
## P.PAKTIKA       0.93483 0.89873 0.97599
## P.PAKTYA        0.97916 0.93997 1.01938
## P.KHOST         1.08601 1.04612 1.13124
## P.KUNARHA       1.17646 1.12875 1.22904
## P.NOORISTAN     1.10876 1.06242 1.15693
## P.BADAKHSHAN    1.06726 1.02092 1.11434
## P.TAKHAR        1.07773 1.03253 1.12671
## P.KUNDUZ        1.14259 1.09479 1.19622
## P.SAMANGAN      0.90234 0.86347 0.94495
## P.BALKH         0.98867 0.94382 1.03264
## P.SAR_E_PUL     0.94393 0.90523 0.98732
## P.GHOR          0.98779 0.94800 1.03668
## P.DAYKUNDI      0.95641 0.91343 1.00133
## P.UROZGAN       1.01925 0.97503 1.06468
## P.ZABUL         1.21152 1.16266 1.26443
## P.KANDAHAR      1.22059 1.17373 1.27114
## P.JAWZJAN       0.98165 0.94055 1.02292
## P.FARYAB        1.03957 0.99736 1.08759
## P.HELMAND       1.27718 1.22568 1.33038
## P.BADGHIS       0.99673 0.95500 1.04305
## P.FARAH         1.18173 1.12846 1.23698
## P.NIMROZ        1.16454 1.11468 1.21848
## area            1.00080 0.98514 1.01623
## other_wives     0.86396 0.85178 0.87556
## HH_sex          0.90688 0.88782 0.92593
```

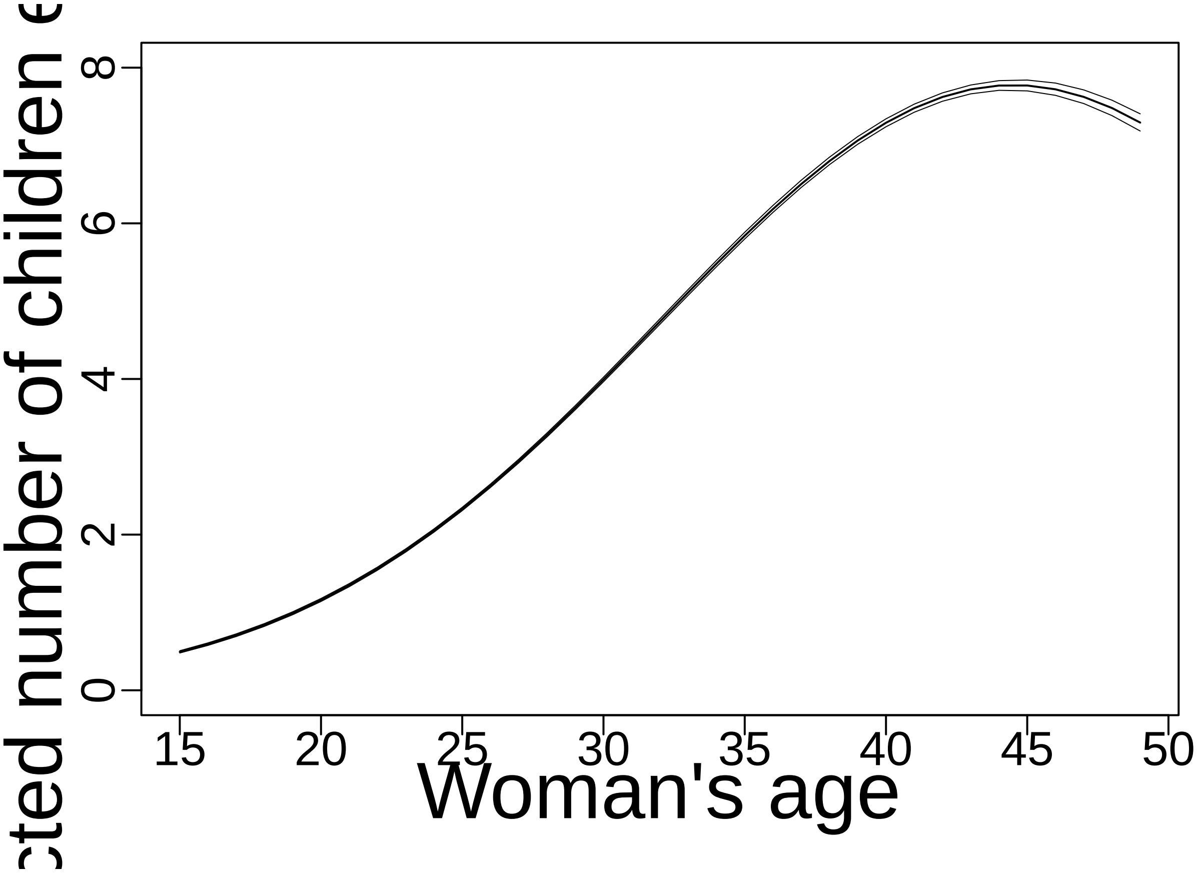
```
sink(file=paste(IMAGEOUT,"CI_THETA.tex",sep=""))
xtable(CI_THETA, digits = 3)
sink()
```

Calculate the expected number of CEB

Expected Children Ever Born - Woman's age

```
#CItheta <- apply(THETA.post, MARGIN=2, function(q){return(quantile(unlist(q),prob=c(0.025,0.5,0.975)))
Xs <- apply(X, 2, mean)
Xs <- matrix(rep(Xs, 35), nrow = 35, byrow = T)
Xs[,1] <- seq(15, 49, by = 1)
Xs[,2] <- Xs[,1]^2
colnames(Xs) <- colnames(X)
Xs = cbind(rep(1,35),Xs) # insert the first column of 1 for intercept
eXB.post <- exp(Xs%*%THETA.post.mat)
qE<-apply(t(eXB.post),2,quantile,probs=c(.025,.5,.975))

sample_size <- NULL
for (v in 15:49){
    sample_size <- c(sample_size, sum(X[,1]==v))
}
qE <- rbind(qE,sample_size)

colnames(qE)<-as.character(15:49)
write.csv(t(qE), paste(IMAGEOUT,"qE.women_age.csv",sep=""))
```

```
par(mar=c(4,4.5,1,1),mgp=c(1.88,0.50,0))
plot( c(15, 49),range(c(0,qE)),type="n",xlab="Woman's age",ylim=c(0,8),
      ylab="Expected number of children ever born", cex.lab = 2.5, cex.axis = 1.5)
lines(seq(15, 49, by = 1), qE[1,],col="black",lwd=0.5)
lines(seq(15, 49, by = 1), qE[2,],col="black",lwd=1.0)
lines(seq(15, 49, by = 1), qE[3,],col="black",lwd=0.5)
```



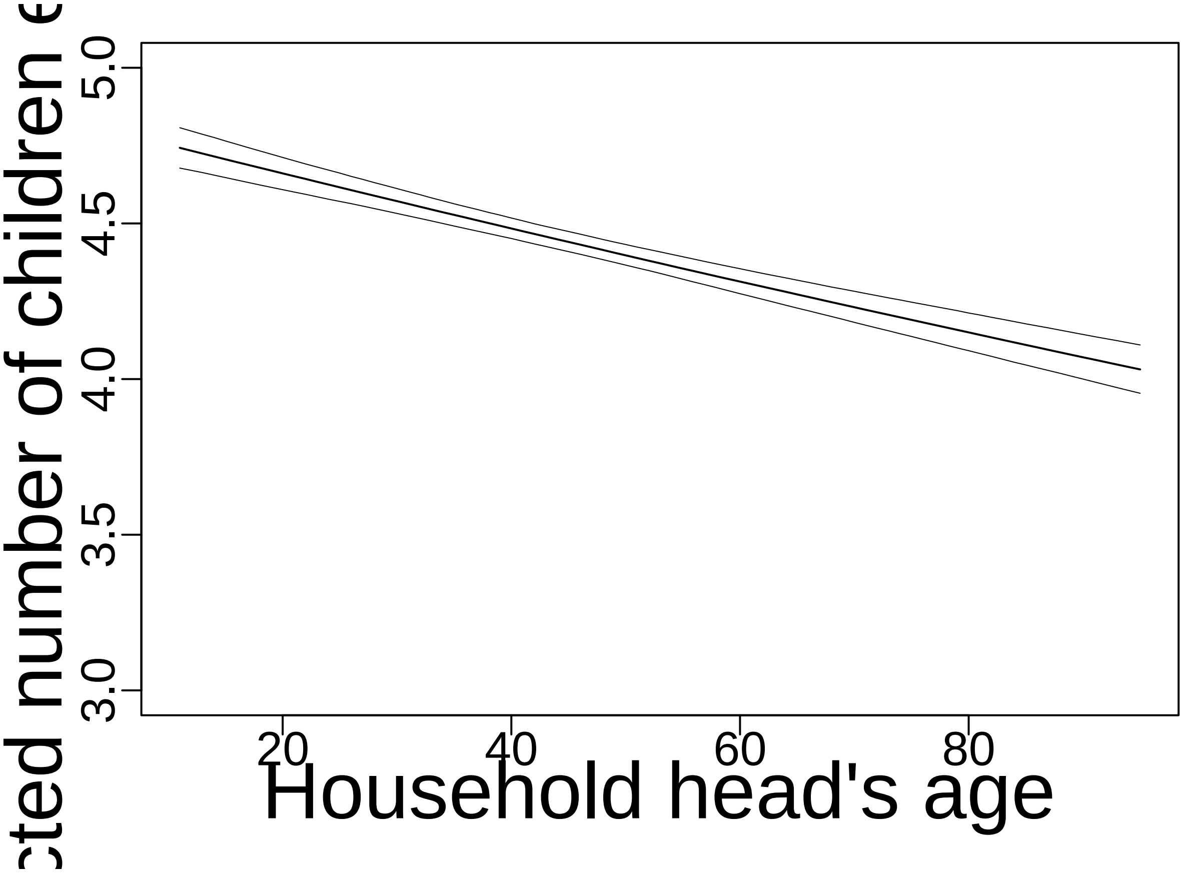
```
dev.print(pdf,
    paste(IMAGEOUT,"E_women_age.pdf",sep=""),
    width=16,height=12)
```

```
## pdf
##   2
```

Expected Children Ever Born - Household Head's age

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- matrix(rep(Xs, 85), nrow = 85, byrow = T)
Xs[,3] <- seq(11, 95, by = 1)
colnames(Xs) <- colnames(X)
Xs = cbind(rep(1,85),Xs) # insert the first column of 1 for intercept
eXB.post <- exp(Xs%*%THETA.post.mat)
qE<-apply(t(eXB.post),2,quantile,probs=c(.025,.5,.975))

sample_size <- NULL
```

```
for (v in 11:95){
    sample_size <- c(sample_size, sum(X[,3]==v))
}
qE <- rbind(qE,sample_size)

colnames(qE)<-as.character(11:95)
write.csv(t(qE), paste(IMAGEOUT,"qE.HH_age.csv",sep=""))

par(mar=c(4,4.5,1,1),mgp=c(1.88,0.50,0))
plot( c(11, 95),range(c(0,qE)),type="n",xlab="Household head's age",ylim=c(3,5),
      ylab="Expected number of children ever born", cex.lab = 2.5, cex.axis = 1.5)
lines(seq(11, 95, by = 1), qE[1,],col="black",lwd=0.5)
lines(seq(11, 95, by = 1), qE[2,],col="black",lwd=1.0)
lines(seq(11, 95, by = 1), qE[3,],col="black",lwd=0.5)
```



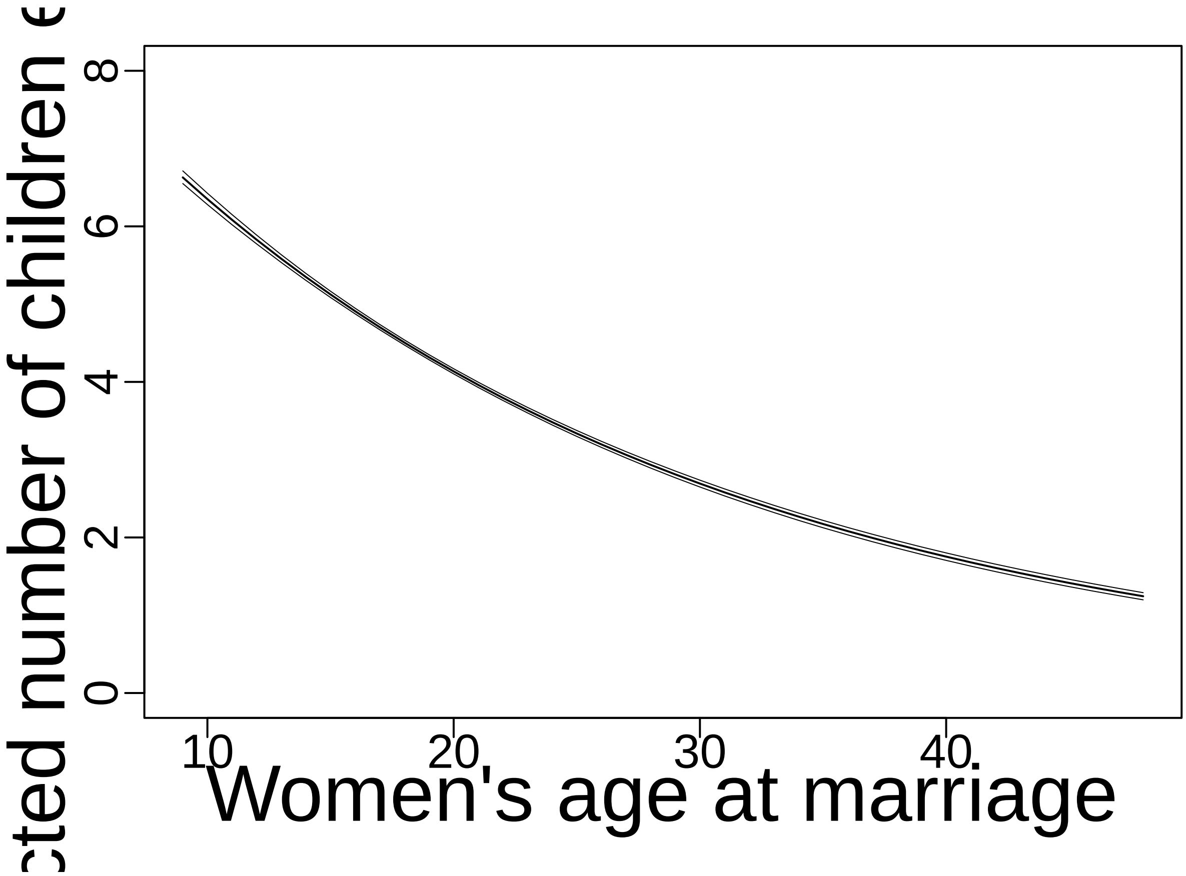
```
dev.print(pdf,
    paste(IMAGEOUT,"E_HH_age.pdf",sep=""),
    width=16,height=12)
```

```
## pdf
##   2
```

Expected Children Ever Born - Woman's age at marriage

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
```

```
Xs <- matrix(rep(Xs, 40), nrow = 40, byrow = T)
Xs[,4] <- seq(9, 48, by = 1)
colnames(Xs) <- colnames(X)
Xs = cbind(rep(1,40),Xs)
eXB.post <- exp(Xs%*%THETA.post.mat)
qE<-apply(t(eXB.post),2,quantile,probs=c(.025,.5,.975))

sample_size <- NULL
for (v in 9:48){
    sample_size <- c(sample_size, sum(X[,5]==v))
}
qE <- rbind(qE,sample_size)

colnames(qE)<-as.character(9:48)
write.csv(t(qE), paste(IMAGEOUT,"qE.women_agem.csv",sep=""))

par(mar=c(4,4.5,1,1),mgp=c(1.88,0.50,0))
plot( c(9, 48),range(c(0,qE)),type="n",xlab="Women's age at marriage",ylim=c(0,8),
      ylab="Expected number of children ever born", cex.lab = 2.5, cex.axis = 1.5)
lines(seq(9, 48, by = 1), qE[1,],col="black",lwd=0.5)
lines(seq(9, 48, by = 1), qE[2,],col="black",lwd=1.0)
lines(seq(9, 48, by = 1), qE[3,],col="black",lwd=0.5)
```



```
dev.print(pdf,
    paste(IMAGEOUT,"E_women_agem.pdf",sep=""),
    width=16,height=12)
```

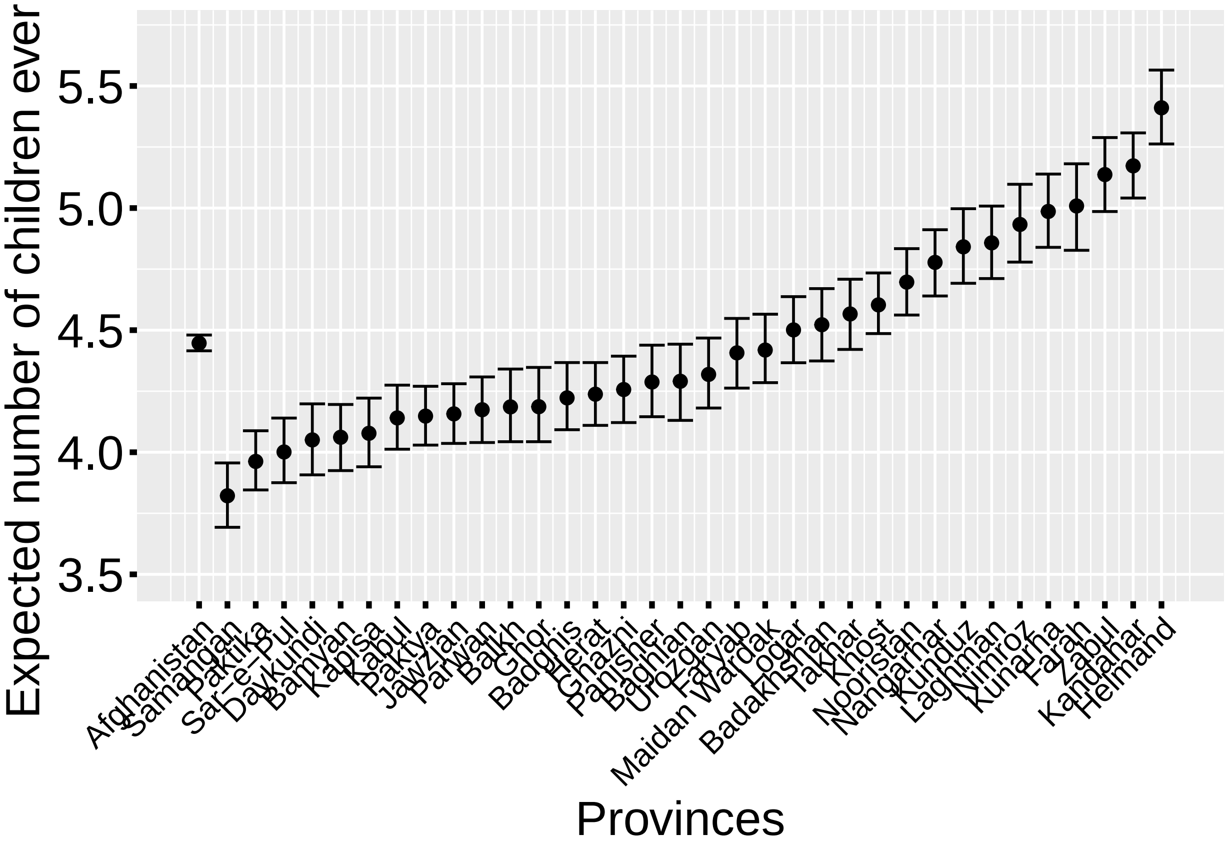
```
## pdf
##   2
```

Expected Children Ever Born - Provinces

```
# Expected Number of Children Ever Born in Afghanistan
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
eXB.post<- exp(t(t(Xs)%*%THETA.post.mat) )
qEA<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))
round(qEA, digits = 3)
```

```
##        [,1]
## 2.5%  4.415
## 50%   4.447
## 97.5% 4.480
```

```
qEA <- rbind(qEA,dim(X)[1])
# Provinces

dnames <- c(
"Afghanistan",
"Herat",
"Kabul",
"Kapisa",
"Parwan",
"Maidan Wardak",
"Logar",
"Nangarhar",
"Laghman",
"Panjsher",
"Baghlan",
"Bamyan",
"Ghazni",
"Paktika",
"Paktya",
"Khost",
"Kunarha",
"Nooristan",
"Badakhshan",
"Takhar",
"Kunduz",
"Samangan",
"Balkh",
"Sar-e-Pul",
"Ghor",
"Daykundi",
"Urozgan",
"Zabul",
"Kandahar",
"Jawzjan",
"Faryab",
"Helmand",
"Badghis",
"Farah",
"Nimroz")
index <- order(dnames)

Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- matrix(rep(Xs, 34), nrow = 34, byrow = T)
for (prov_index in c(2:34)){
    x <- rep(0,34)
    x[prov_index] <- 1
    Xs[,(8+prov_index)] <- x
}
colnames(Xs) <- colnames(X)
Xs = cbind(rep(1,34),Xs)

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- sum(df_sel$province==32) # count number of women in Herat(reference levels)
for (colidx in 11:43){
    sample_size <- c(sample_size, sum(X[,colidx]==1))
}
qE <- rbind(qE,sample_size)

qE.out <- cbind(qEA,qE)
colnames(qE.out)<-dnames
write.csv(t(qE.out), paste(IMAGEOUT,"qE.province.csv",sep=""))

index <- order(c(qEA[2,], qE[2,])[-1])
index <- c(1, index + 1)

library(ggplot2)
BHP.df_province <- data.frame(x =1:35,
                 F = c(qEA[2,], qE[2,])[index],
                 L = c(qEA[1,], qE[1,])[index],
                 U = c(qEA[3,], qE[3,])[index],
              name = dnames[index])
abc <- ggplot(BHP.df_province, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Provinces") +
  ylim(3.5,5.7) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:35), labels = dnames[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
save(BHP.df_province, file=paste(IMAGEOUT,"BHP.df_province.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_province.pdf",sep=""),width=8,height=8)
```

```
## pdf
##   2
```

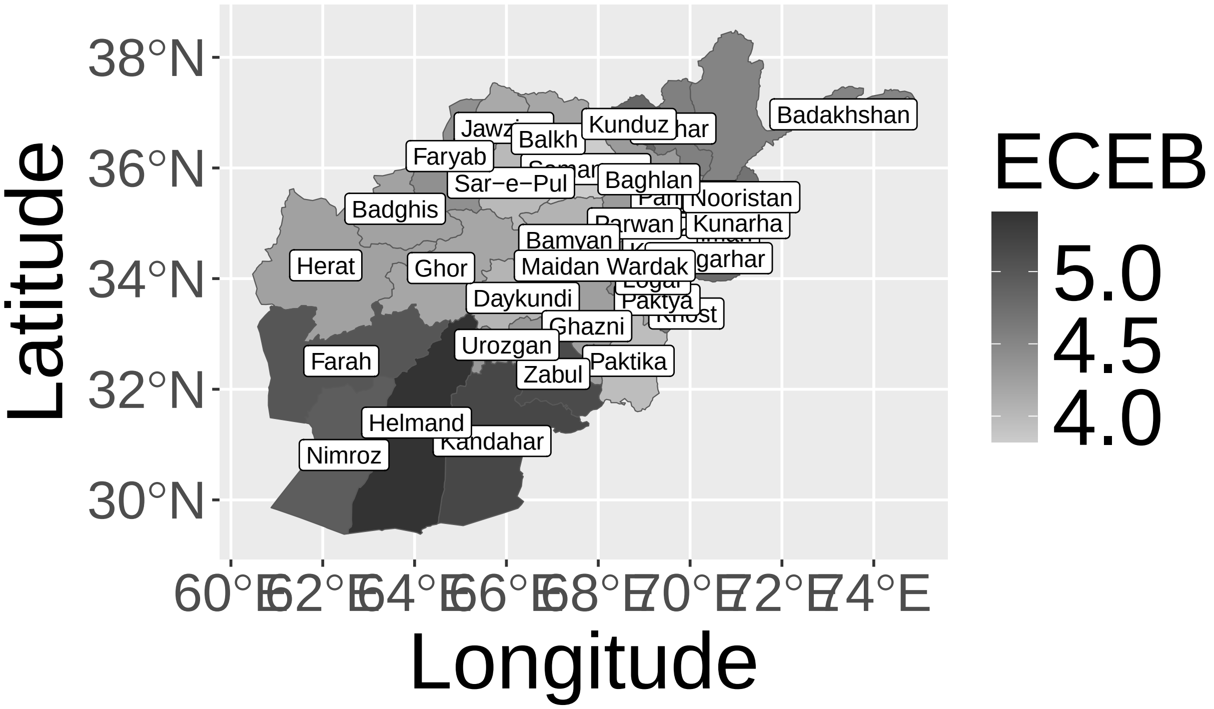
```
abc <- ggplot(df[-1,], aes(x = x-1, y = U-L)) +
  scale_x_continuous(breaks = c(1:34), labels = dnames[index[-1]]) +
  labs(y = "Width of CI of expected CEB", x = "Province") +
  ylim(0,0.4) +
geom_bar(stat="identity", position="dodge", width=0.5)+
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
dev.print(pdf, paste(IMAGEOUT, "CI.Width.E_province.pdf",sep=""),width=8,height=8)
print("variance of the width of CI: ")
print(var(qE[3,]-qE[1,]))
```

```
library(ggplot2)
library(sf)

map <- read_sf("data_src/map.json")
eCEB <- NULL
for (n in map$name){
    print(n)
    print(which(BHP.df_province$name==n))
    print(BHP.df_province$name[which(BHP.df_province$name==n)])
    print(BHP.df_province$F[which(BHP.df_province$name==n)])
    eCEB <- append(eCEB, BHP.df_province$F[which(BHP.df_province$name==n)])
}
```

```
## [1] "Kapisa"
## [1] 7
## [1] "Kapisa"
## [1] 4.077586
## [1] "Panjsher"
## [1] 17
## [1] "Panjsher"
## [1] 4.287572
## [1] "Laghman"
## [1] 29
## [1] "Laghman"
## [1] 4.857453
## [1] "Khost"
## [1] 25
## [1] "Khost"
## [1] 4.603444
## [1] "Kabul"
## [1] 8
## [1] "Kabul"
## [1] 4.14059
## [1] "Kunarha"
## [1] 31
## [1] "Kunarha"
## [1] 4.986141
## [1] "Paktya"
## [1] 9
## [1] "Paktya"
## [1] 4.148028
## [1] "Parwan"
## [1] 11
## [1] "Parwan"
## [1] 4.174068
## [1] "Nooristan"
## [1] 26
## [1] "Nooristan"
## [1] 4.696769
## [1] "Jawzjan"
## [1] 10
## [1] "Jawzjan"
## [1] 4.157181
## [1] "Takhar"
## [1] 24
## [1] "Takhar"
## [1] 4.566279
## [1] "Samangan"
## [1] 2
## [1] "Samangan"
## [1] 3.821736
## [1] "Sar-e-Pul"
## [1] 4
## [1] "Sar-e-Pul"
## [1] 4.001142
## [1] "Daykundi"
## [1] 5
## [1] "Daykundi"
## [1] 4.050481
## [1] "Balkh"
## [1] 12
## [1] "Balkh"
## [1] 4.185731
## [1] "Baghlan"
## [1] 18
## [1] "Baghlan"
## [1] 4.290693
## [1] "Bamyan"
## [1] 6
## [1] "Bamyan"
## [1] 4.061326
## [1] "Paktika"
## [1] 3
## [1] "Paktika"
## [1] 3.962403
## [1] "Faryab"
## [1] 20
## [1] "Faryab"
## [1] 4.407024
## [1] "Ghazni"
## [1] 16
## [1] "Ghazni"
## [1] 4.256818
## [1] "Ghor"
## [1] 13
## [1] "Ghor"
## [1] 4.18657
## [1] "Nimroz"
## [1] 30
## [1] "Nimroz"
## [1] 4.93287
## [1] "Herat"
## [1] 15
## [1] "Herat"
## [1] 4.237754
## [1] "Kandahar"
## [1] 34
## [1] "Kandahar"
## [1] 5.172886
## [1] "Kunduz"
## [1] 28
## [1] "Kunduz"
## [1] 4.841213
## [1] "Helmand"
## [1] 35
## [1] "Helmand"
## [1] 5.410587
## [1] "Badghis"
## [1] 14
## [1] "Badghis"
## [1] 4.222639
## [1] "Zabul"
## [1] 33
## [1] "Zabul"
## [1] 5.13713
## [1] "Urozgan"
## [1] 19
## [1] "Urozgan"
## [1] 4.318706
## [1] "Farah"
## [1] 32
## [1] "Farah"
## [1] 5.008579
## [1] "Badakhshan"
## [1] 23
## [1] "Badakhshan"
## [1] 4.522184
## [1] "Nangarhar"
## [1] 27
## [1] "Nangarhar"
## [1] 4.77746
## [1] "Logar"
## [1] 22
## [1] "Logar"
## [1] 4.50099
## [1] "Maidan Wardak"
## [1] 21
## [1] "Maidan Wardak"
## [1] 4.418495
```

```
map$eCEB <- eCEB
ggplot(map) +
    geom_sf(aes(fill = eCEB)) +
    scale_fill_gradient(low="gray80", high="gray20") +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill="ECEB", x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 30),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```



```
dev.print(pdf, paste(IMAGEOUT, "E_province_map.pdf",sep=""),width=16,height=10)
```

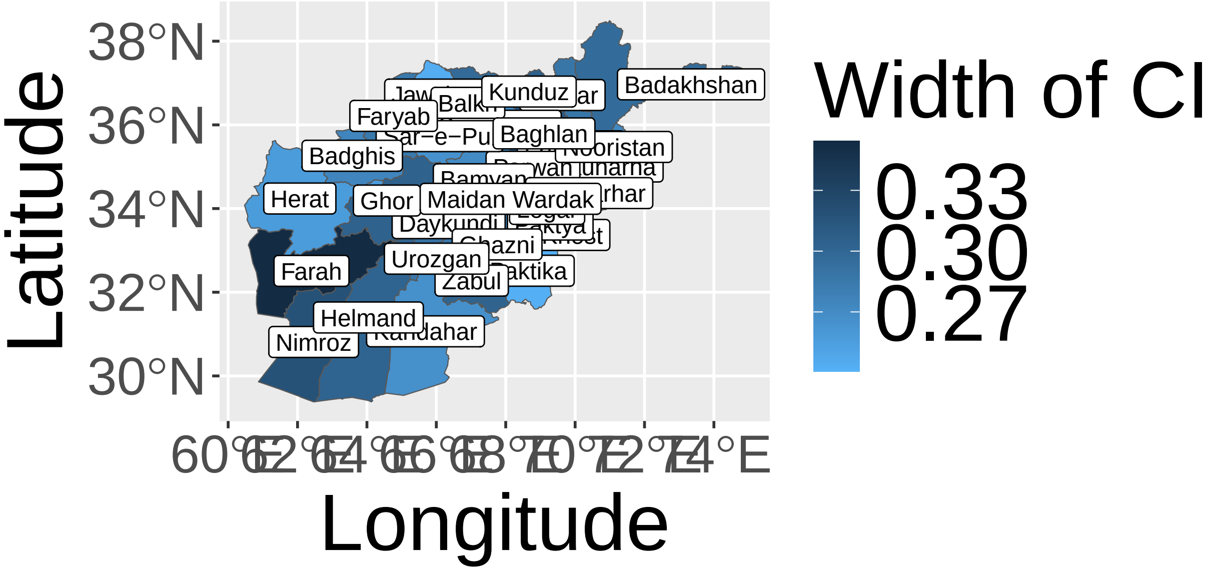
```
## pdf
##   2
```

```
library(ggplot2)
library(sf)

map <- read_sf("data_src/map.json")
eCEB.CI.width <- NULL
for (n in map$name){
    print(n)
    print(which(BHP.df_province$name==n))
    print(BHP.df_province$name[which(BHP.df_province$name==n)])
    print(BHP.df_province$F[which(BHP.df_province$name==n)])
    eCEB.CI.width <- append(eCEB.CI.width, BHP.df_province$U[which(BHP.df_province$name==n)]-BHP.df_pro
}
```

```
## [1] "Kapisa"
## [1] 7
## [1] "Kapisa"
## [1] 4.077586
## [1] "Panjsher"
## [1] 17
## [1] "Panjsher"
## [1] 4.287572
## [1] "Laghman"
## [1] 29
## [1] "Laghman"
## [1] 4.857453
## [1] "Khost"
## [1] 25
## [1] "Khost"
## [1] 4.603444
## [1] "Kabul"
## [1] 8
## [1] "Kabul"
## [1] 4.14059
## [1] "Kunarha"
## [1] 31
## [1] "Kunarha"
## [1] 4.986141
## [1] "Paktya"
## [1] 9
## [1] "Paktya"
## [1] 4.148028
## [1] "Parwan"
## [1] 11
## [1] "Parwan"
## [1] 4.174068
## [1] "Nooristan"
## [1] 26
## [1] "Nooristan"
## [1] 4.696769
## [1] "Jawzjan"
## [1] 10
## [1] "Jawzjan"
## [1] 4.157181
## [1] "Takhar"
## [1] 24
## [1] "Takhar"
## [1] 4.566279
## [1] "Samangan"
## [1] 2
## [1] "Samangan"
## [1] 3.821736
## [1] "Sar-e-Pul"
## [1] 4
## [1] "Sar-e-Pul"
## [1] 4.001142
## [1] "Daykundi"
## [1] 5
## [1] "Daykundi"
## [1] 4.050481
## [1] "Balkh"
## [1] 12
## [1] "Balkh"
## [1] 4.185731
## [1] "Baghlan"
## [1] 18
## [1] "Baghlan"
## [1] 4.290693
## [1] "Bamyan"
## [1] 6
## [1] "Bamyan"
## [1] 4.061326
## [1] "Paktika"
## [1] 3
## [1] "Paktika"
## [1] 3.962403
## [1] "Faryab"
## [1] 20
## [1] "Faryab"
## [1] 4.407024
## [1] "Ghazni"
## [1] 16
## [1] "Ghazni"
## [1] 4.256818
## [1] "Ghor"
## [1] 13
## [1] "Ghor"
## [1] 4.18657
## [1] "Nimroz"
## [1] 30
## [1] "Nimroz"
## [1] 4.93287
## [1] "Herat"
## [1] 15
## [1] "Herat"
## [1] 4.237754
## [1] "Kandahar"
## [1] 34
## [1] "Kandahar"
## [1] 5.172886
## [1] "Kunduz"
## [1] 28
## [1] "Kunduz"
## [1] 4.841213
## [1] "Helmand"
## [1] 35
## [1] "Helmand"
## [1] 5.410587
## [1] "Badghis"
## [1] 14
## [1] "Badghis"
## [1] 4.222639
## [1] "Zabul"
## [1] 33
## [1] "Zabul"
## [1] 5.13713
## [1] "Urozgan"
## [1] 19
## [1] "Urozgan"
## [1] 4.318706
## [1] "Farah"
## [1] 32
## [1] "Farah"
## [1] 5.008579
## [1] "Badakhshan"
## [1] 23
## [1] "Badakhshan"
## [1] 4.522184
## [1] "Nangarhar"
## [1] 27
## [1] "Nangarhar"
## [1] 4.77746
## [1] "Logar"
## [1] 22
## [1] "Logar"
## [1] 4.50099
## [1] "Maidan Wardak"
## [1] 21
## [1] "Maidan Wardak"
## [1] 4.418495
```

```
map$eCEB.CI.width <- eCEB.CI.width
ggplot(map) +
    geom_sf(aes(fill = eCEB.CI.width)) +
    scale_fill_gradientn(colors = c("#56B1F7", "#132B43"),
      values = scales::rescale(c(0.15,0.33)),
      guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill="Width of CI", x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 30),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```



```
dev.print(pdf, paste(IMAGEOUT, "CI.Width.E_province_map.pdf",sep=""),width=16,height=10)
```

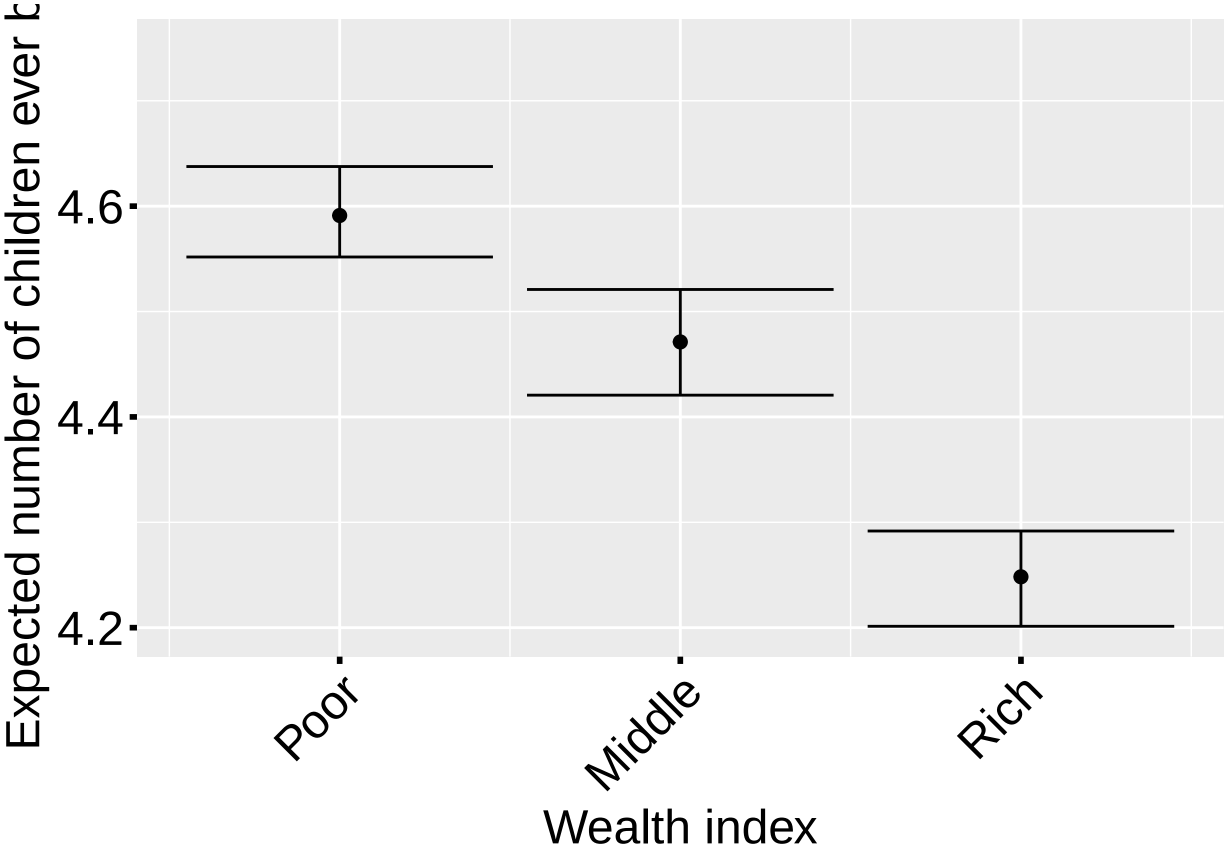
```
## pdf
##   2
```

Expected Children Ever Born - Wealth Index

```
winames <- c("Middle", "Poor", "Rich")
index <- c(2, 1, 3)
winames[index]
```

```
## [1] "Poor"   "Middle" "Rich"
```

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 3), nrow = 3, byrow = T)
Xs[,8] <- c(0, 1, 0)
Xs[,9] <- c(0, 0, 1)

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- sum(df_sel$windex==3) # count number of women with "Middle" wealth level
for (colidx in 8:9){
    sample_size <- c(sample_size, sum(X[,colidx]==1))
}
qE <- rbind(qE,sample_size)

colnames(qE)<-winames
write.csv(t(qE), paste(IMAGEOUT,"qE.windex.csv",sep=""))

BHP.df_windex3 <- data.frame(x =1:3,
                 F = c(qE[2,])[index],
                 L = c(qE[1,])[index],
                 U = c(qE[3,])[index])
abc <- ggplot(BHP.df_windex3, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Wealth index", cex.lab = 2.5, cex.axis = 1.5) +
  geom_point(size = 2) +
  ylim(4.2,4.75) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:3), labels = winames[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
save(BHP.df_windex3, file=paste(IMAGEOUT,"BHP.df_windex3.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_windex3.pdf",sep=""),width=8,height=8)
```

```
## pdf
##   2
```

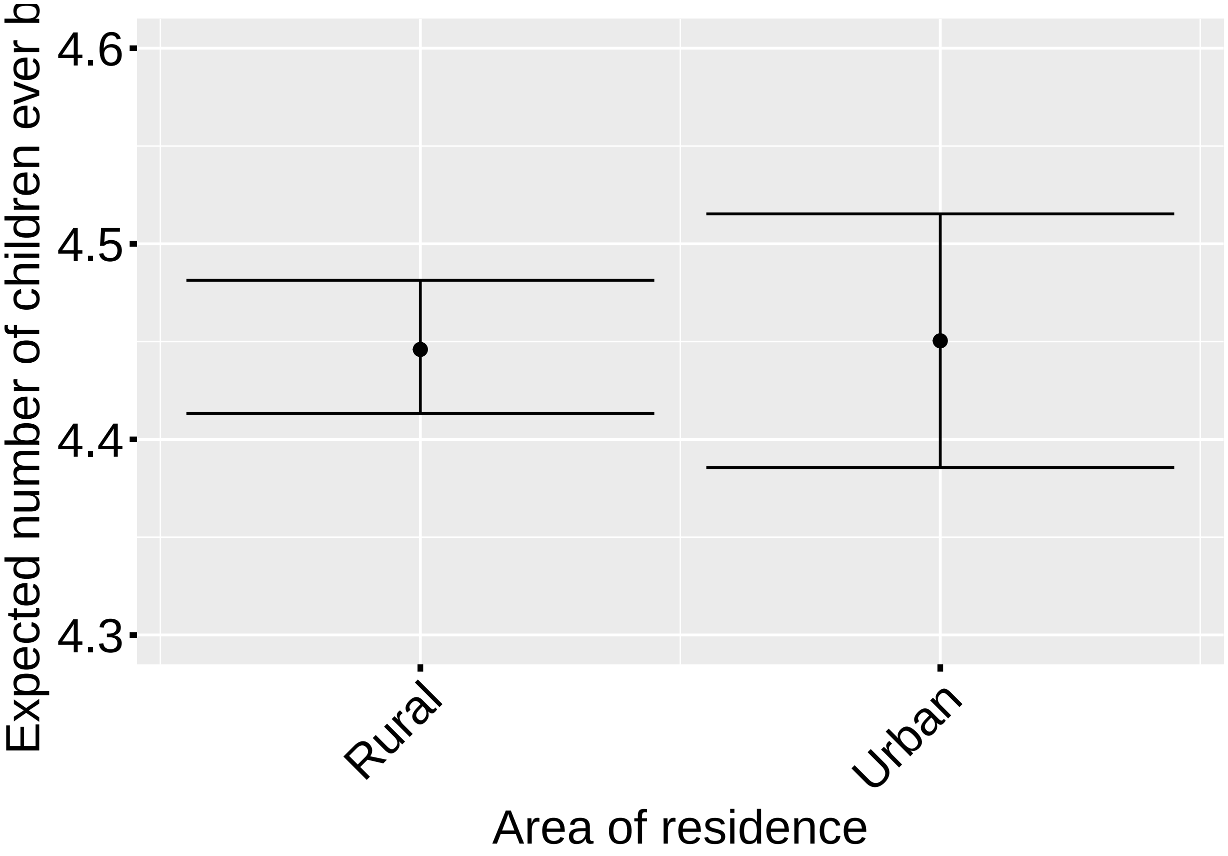
Expected Children Ever Born - Area

```
# To check the X value for area, 2 for Urban, 1 for Rural
# (it is flipped because of refactor)
# unique(as.numeric(area)-X[,44])

anames<- c("Rural", "Urban")
index <- c(1, 2)
anames[index]
```

```
## [1] "Rural" "Urban"
```

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 2), nrow = 2, byrow = T)
Xs[,44] <- c(1, 2) # 2 for Urban, 1 for Rural

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- c(sum(X[,44]==1), sum(X[,44]==2)) # Rural, Urbann
qE <- rbind(qE,sample_size)
```

```r
colnames(qE)<-anames
write.csv(t(qE), paste(IMAGEOUT,"qE.area.csv",sep=""))

BHP.df_area <- data.frame(x =1:2,
                 F = c(qE[2,])[index],
                 L = c(qE[1,])[index],
                 U = c(qE[3,])[index])
abc <- ggplot(BHP.df_area, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Area of residence") +
  geom_point(size = 2) +
  ylim(4.3,4.6) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:2), labels = anames[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```r
save(BHP.df_area, file=paste(IMAGEOUT,"BHP.df_area.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_area.pdf",sep=""),width=8,height=8)
```

```
## pdf
##   2
```

```
(qE[2,][1] - qE[2,][2])/qE[2,][2]
```

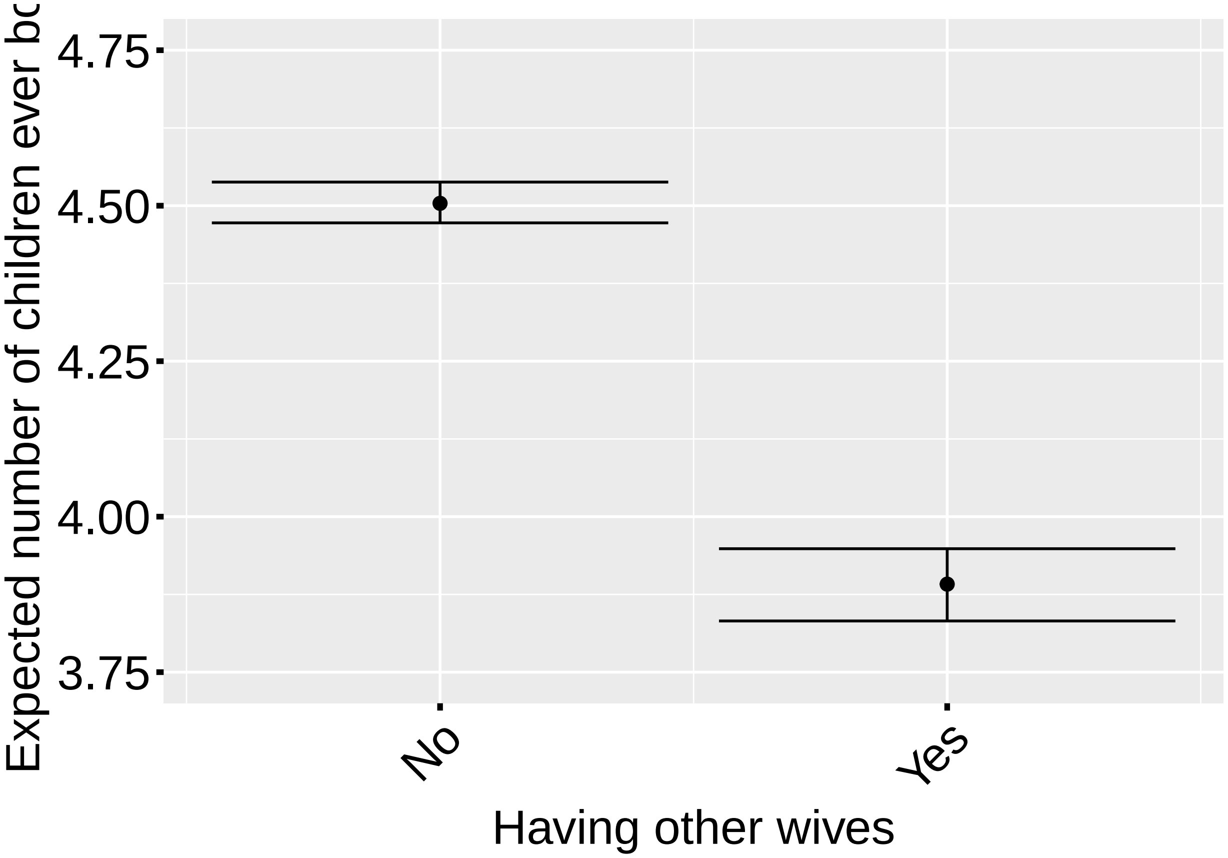
```
##        Rural
## -0.000991774
```

Expected Children Ever Born - Have other wives

```
otherwives_labels <- c("No", "Yes")
index <- c(1, 2)
otherwives_labels[index]
```

```
## [1] "No"  "Yes"
```

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 2), nrow = 2, byrow = T)
Xs[,45] <- c(1, 2) # 1 for No other wives, 2 for one or more

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- c(sum(X[,45]==1), sum(X[,45]==2)) # No, Yes
qE <- rbind(qE,sample_size)

colnames(qE)<-otherwives_labels
write.csv(t(qE), paste(IMAGEOUT,"qE.other_wives.csv",sep=""))

BHP.df_other_wives <- data.frame(x =1:2,
                 F = c(qE[2,])[index],
                 L = c(qE[1,])[index],
                 U = c(qE[3,])[index])
abc <- ggplot(BHP.df_other_wives, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Having other wives") +
  geom_point(size = 2) +
  ylim(3.75,4.75) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:2), labels = otherwives_labels[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```r
save(BHP.df_other_wives, file=paste(IMAGEOUT,"BHP.df_other_wives.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_other_wives.pdf",sep=""),width=8,height=8)
```

```
## pdf
##   2
```

```r
(qE[2,][1] - qE[2,][2])/qE[2,][2]
```

```
##        No
## 0.1573449
```

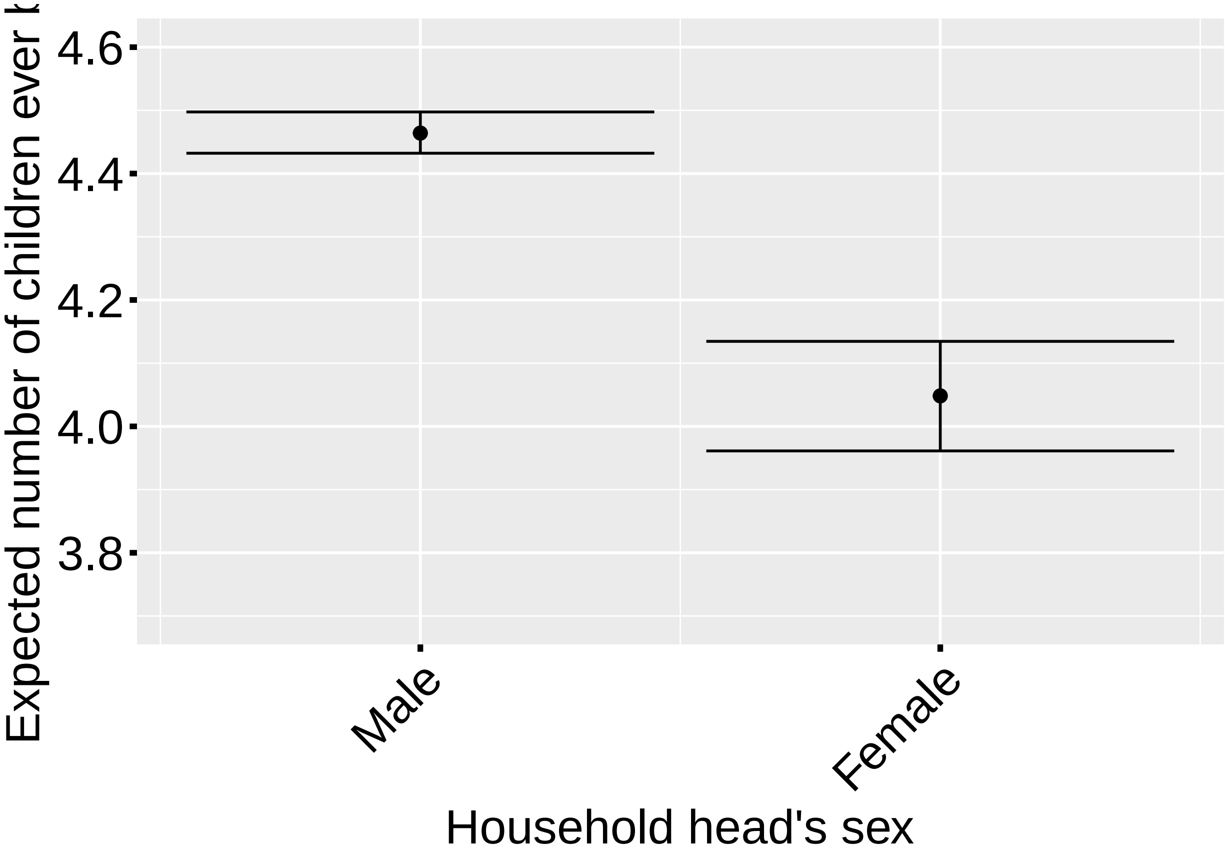
Expected Children Ever Born - Household Head sex

```r
HH_sex_labels <- c("Male", "Female")
index <- c(1, 2)
HH_sex_labels[index]
```

```
## [1] "Male"   "Female"
```

```r
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 2), nrow = 2, byrow = T)
Xs[,46] <- c(1, 2) # 1 for Male, 2 for Female

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- c(sum(X[,45]==1), sum(X[,45]==2)) # Male, Female
```

```r
qE <- rbind(qE,sample_size)

colnames(qE)<-HH_sex_labels
write.csv(t(qE), paste(IMAGEOUT,"qE.HH_sex.csv",sep=""))

BHP.df_HH_sex <- data.frame(x =1:2,
                  F = c(qE[2,])[index],
                  L = c(qE[1,])[index],
                  U = c(qE[3,])[index])
abc <- ggplot(BHP.df_HH_sex, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Household head's sex") +
  geom_point(size = 2) +
  ylim(3.7,4.6) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:2), labels = HH_sex_labels[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```r
save(BHP.df_HH_sex, file=paste(IMAGEOUT,"BHP.df_HH_sex.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_HH_sex.pdf",sep=""),width=8,height=8)
```

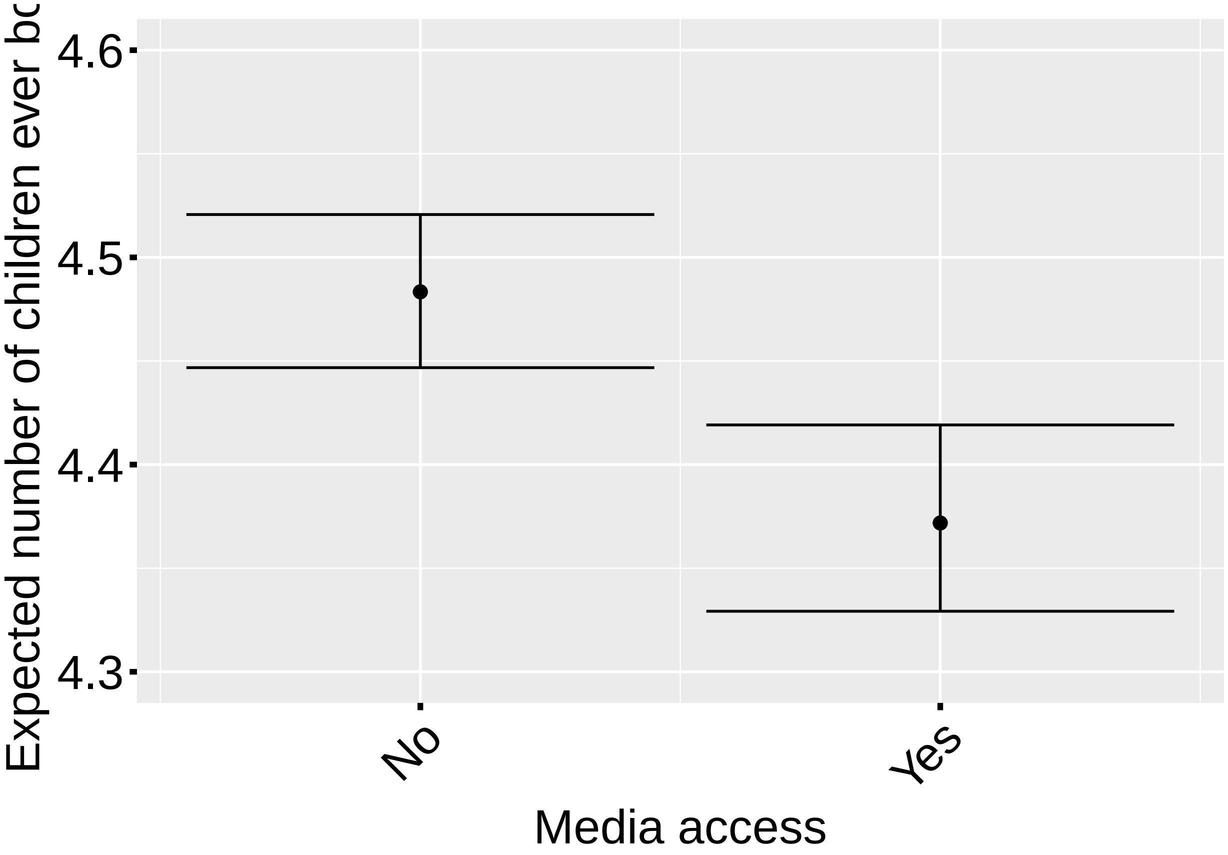
```
## pdf
##   2
```

Expected Children Ever Born - Media exposure

```
media_labels <- c("No", "Yes")
index <- c(1, 2)
media_labels[index]
```

```
## [1] "No"  "Yes"
```

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 2), nrow = 2, byrow = T)
Xs[,10] <- c(1, 2) # 1 for No, 2 for Yes

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- c(sum(X[,9]==1), sum(X[,9]==2)) # No, Yes
qE <- rbind(qE,sample_size)

colnames(qE)<-media_labels
write.csv(t(qE), paste(IMAGEOUT,"qE.media.csv",sep=""))

BHP.df_media <- data.frame(x =1:2,
                 F = c(qE[2,])[index],
                 L = c(qE[1,])[index],
                 U = c(qE[3,])[index])
abc <- ggplot(BHP.df_media, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Media access") +
  geom_point(size = 2) +
  ylim(4.3,4.6) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:2), labels = media_labels[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```r
save(BHP.df_media, file=paste(IMAGEOUT,"BHP.df_media.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_media.pdf",sep=""),width=8,height=8)
```

```
## pdf
##   2
```

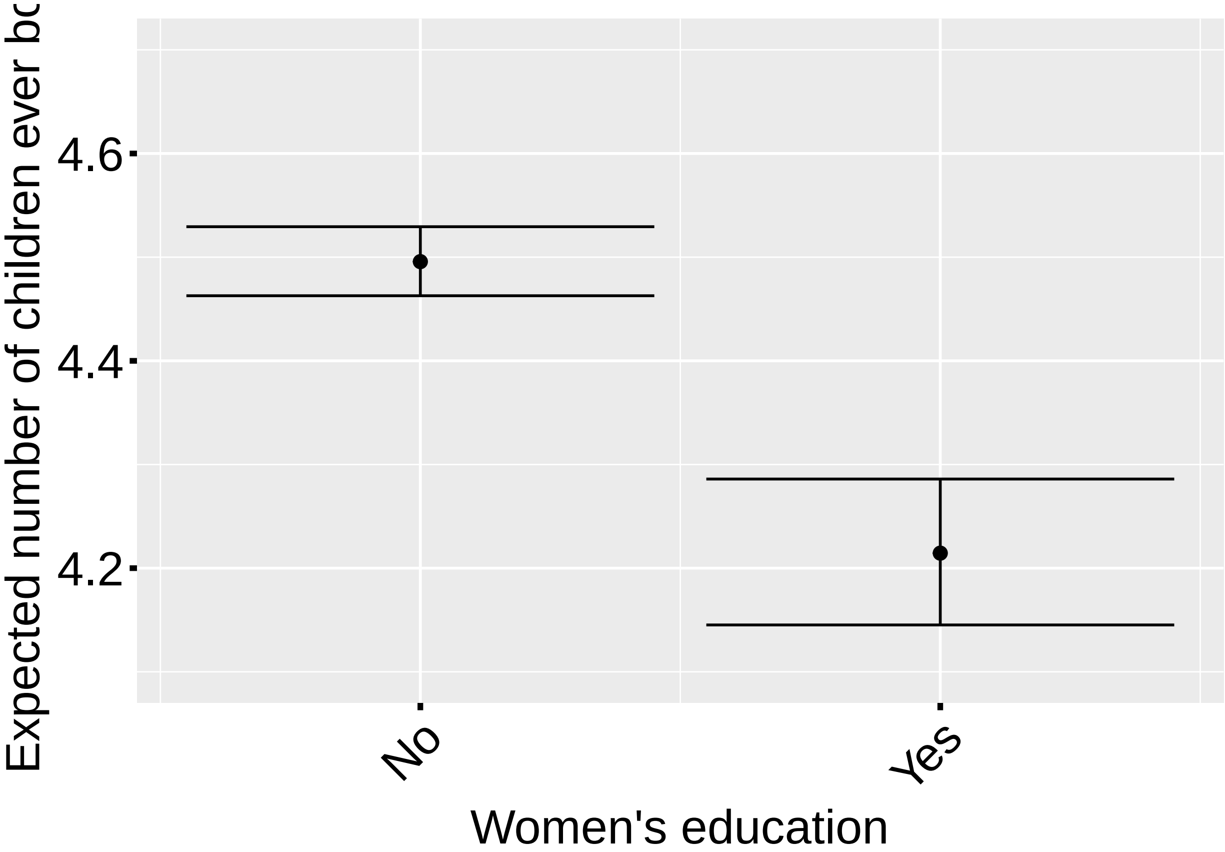
Expected Children Ever Born - Woman's education

```r
women_edu_labels <- c("No", "Yes")
index <- c(1, 2)
women_edu[index]
```

```
## [1] 1 0
## Levels: 0 1
```

```r
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 2), nrow = 2, byrow = T)
Xs[,6] <- c(1, 2) # 1 for No, 2 for Yes

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- c(sum(X[,5]==1), sum(X[,5]==2)) # No, Yes
qE <- rbind(qE,sample_size)

colnames(qE)<-women_edu_labels
write.csv(t(qE), paste(IMAGEOUT,"qE.women_edu.csv",sep=""))
```

```
BHP.df_women_edu <- data.frame(x =1:2,
                  F = c(qE[2,])[index],
                  L = c(qE[1,])[index],
                  U = c(qE[3,])[index])
abc <- ggplot(BHP.df_women_edu, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Women's education") +
  geom_point(size = 2) +
  ylim(4.1,4.7) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:2), labels = women_edu_labels[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
save(BHP.df_women_edu, file=paste(IMAGEOUT,"BHP.df_women_edu.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_women_edu.pdf",sep=""),width=8,height=8)
```

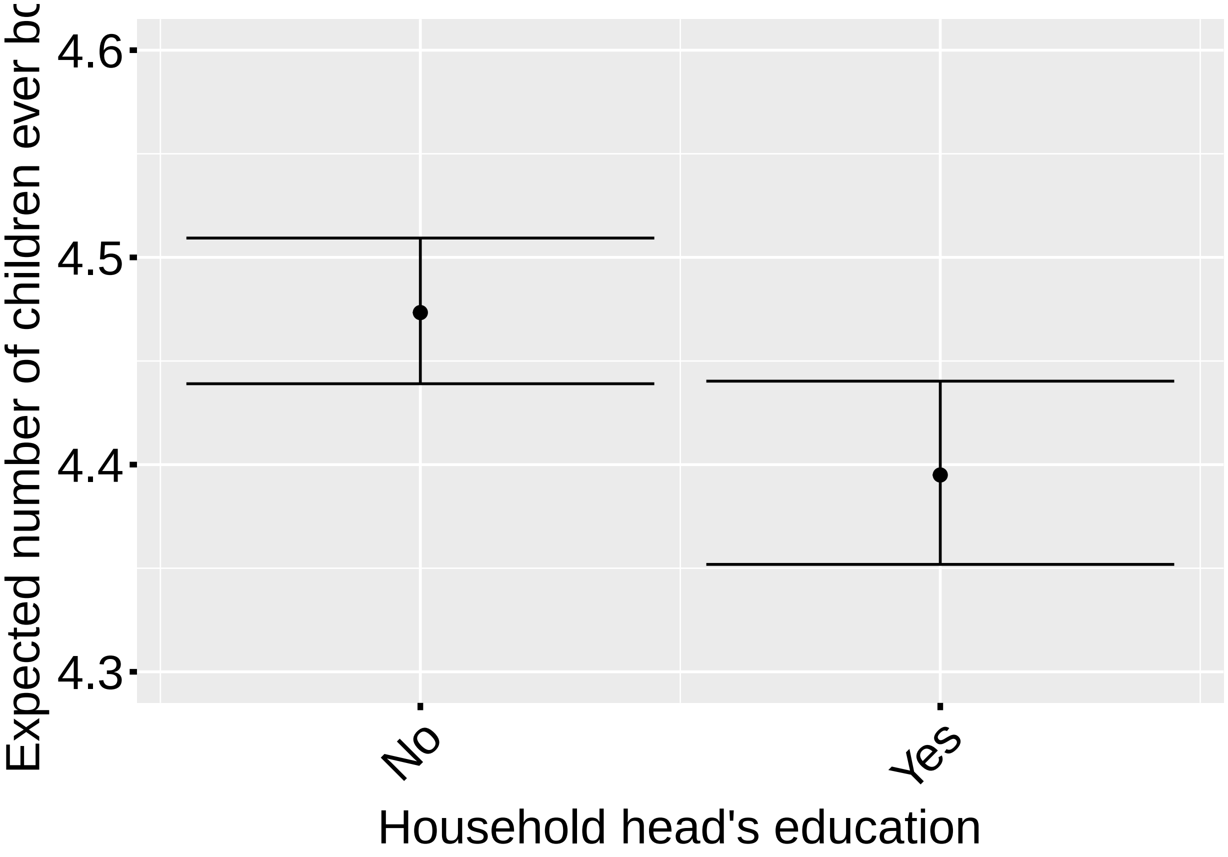
```
## pdf
##   2
```

Expected Children Ever Born - Household Head’s education

```
HH_edu_labels <- c("No", "Yes")
index <- c(1, 2)
HH_edu_labels[index]
```

```
## [1] "No"  "Yes"
```

```
Xs <- apply(X, 2, mean)
Xs[2] <- Xs[1]^2
Xs <- as.matrix(c(1, Xs))
Xs <- matrix(rep(Xs, 2), nrow = 2, byrow = T)
Xs[,7] <- c(1, 2) # 1 for No, 2 for Yes

eXB.post<- exp(t(Xs%*%THETA.post.mat) )
qE<-apply( eXB.post,2,quantile,probs=c(.025,.5,.975))

sample_size <- c(sum(X[,6]==1), sum(X[,6]==2)) # No, Yes
qE <- rbind(qE,sample_size)

colnames(qE)<-HH_edu_labels
write.csv(t(qE), paste(IMAGEOUT,"qE.HH_edu.csv",sep=""))

BHP.df_HH_edu <- data.frame(x =1:2,
                 F = c(qE[2,])[index],
                 L = c(qE[1,])[index],
                 U = c(qE[3,])[index])
abc <- ggplot(BHP.df_HH_edu, aes(x = x, y = F)) +
  labs(y = "Expected number of children ever born", x = "Household head's education") +
  geom_point(size = 2) +
  ylim(4.3,4.6)+
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:2), labels = HH_edu_labels[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 18, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



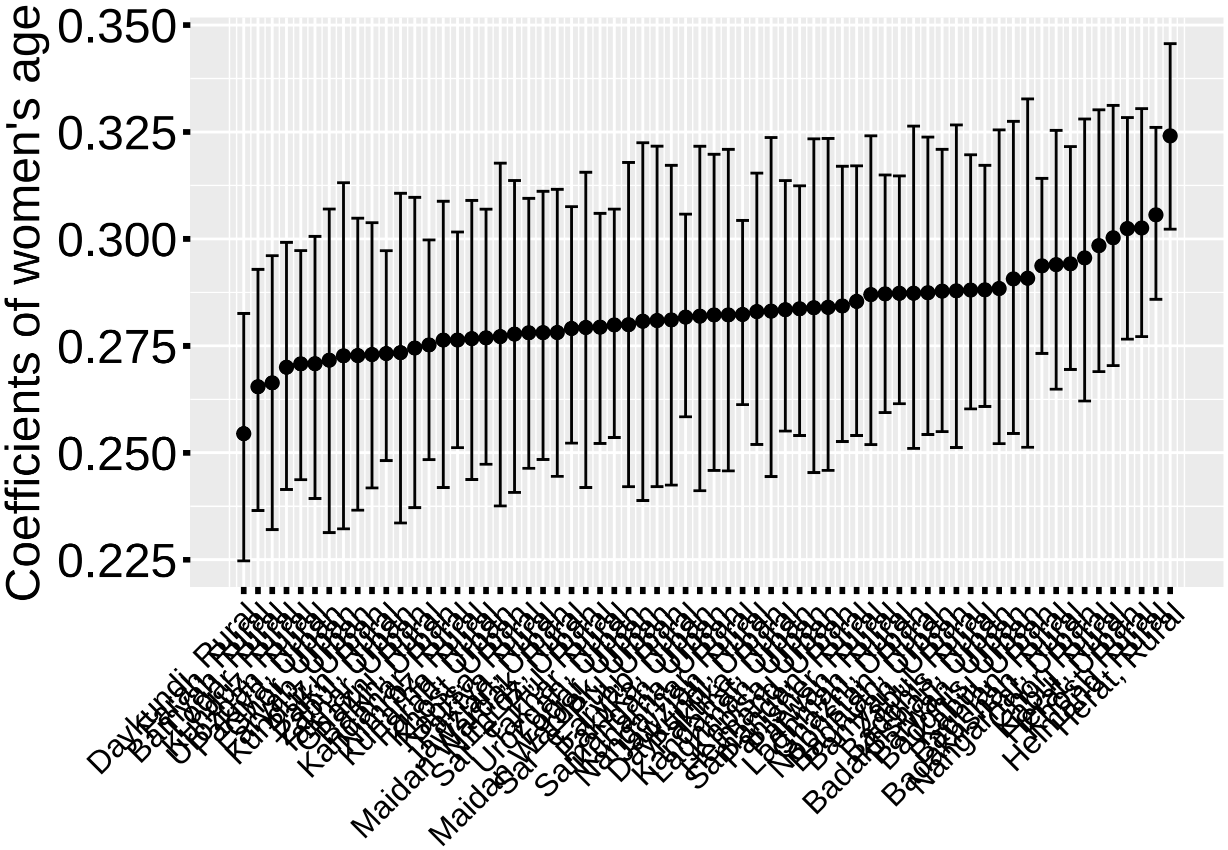
```r
save(BHP.df_HH_edu, file=paste(IMAGEOUT,"BHP.df_HH_edu.RData",sep=""))
dev.print(pdf, paste(IMAGEOUT, "E_HH_edu.pdf",sep=""),width=8,height=8)
```

```
## pdf
##   2
```

Posterior CI of GAMMA + THETA

To plot the posterior CI of gamma + theta coefficients of women_age

```r
# stratum_vector stores the labels of all 66 stratum
#stratum_vector <- unique(df_sel$stratum)
i <- 1
GAMMA_women_age <- NULL
stratum_label <- NULL
stratum_area <- NULL
stratum_province <- NULL
stratum_sample_size <- NULL
for (stratum in stratum_vector){
    stratum_str <- toString(stratum)
    stratum_sample_size <- c(stratum_sample_size, sum(df_sel$stratum==stratum))
    area <- substr(stratum_str, nchar(stratum_str), nchar(stratum_str))
    area <- case_when(
        (area == "1") ~ "Urban",
        (area == "2") ~ "Rural"
    )
    province <- substr(stratum_str, 1, nchar(stratum_str)-1)
    province <- case_when(
        (province == "1") ~ "Kabul",
        (province == "2") ~ "Kapisa",
        (province == "3") ~ "Parwan",
        (province == "4") ~ "Maidan Wardak",
        (province == "5") ~ "Logar",
        (province == "6") ~ "Nangarhar",
        (province == "7") ~ "Laghman",
        (province == "8") ~ "Panjsher",
        (province == "9") ~ "Baghlan",
        (province == "10") ~ "Bamyan",
        (province == "11") ~ "Ghazni",
        (province == "12") ~ "Paktika",
        (province == "13") ~ "Paktya",
        (province == "14") ~ "Khost",
        (province == "15") ~ "Kunarha",
        (province == "16") ~ "Nooristan",
        (province == "17") ~ "Badakhshan",
        (province == "18") ~ "Takhar",
        (province == "19") ~ "Kunduz",
        (province == "20") ~ "Samangan",
        (province == "21") ~ "Balkh",
        (province == "22") ~ "Sar-e-Pul",
        (province == "23") ~ "Ghor",
        (province == "24") ~ "Daykundi",
        (province == "25") ~ "Urozgan",
        (province == "26") ~ "Zabul",
        (province == "27") ~ "Kandahar",
        (province == "28") ~ "Jawzjan",
        (province == "29") ~ "Faryab",
        (province == "30") ~ "Helmand",
        (province == "31") ~ "Badghis",
        (province == "32") ~ "Herat",
        (province == "33") ~ "Farah",
        (province == "34") ~ "Nimroz"
    )
    stratum_label <- c(stratum_label, paste(province,", ",area,sep=""))
    stratum_area<- c(stratum_area, area)
    stratum_province<- c(stratum_province, province)

    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 2])
    }
    GAMMA_women_age <- rbind(GAMMA_women_age, gamma_ij)
    i <- i+1
}
# THETA.post.mat[2,] is the fixed effect of fixed predictor
qE<-apply(GAMMA_women_age+rep(THETA.post.mat[2,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of women's age", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:m), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
dev.print(pdf, paste(IMAGEOUT, "BETA.women_age.pdf",sep=""),width=16,height=8)
```

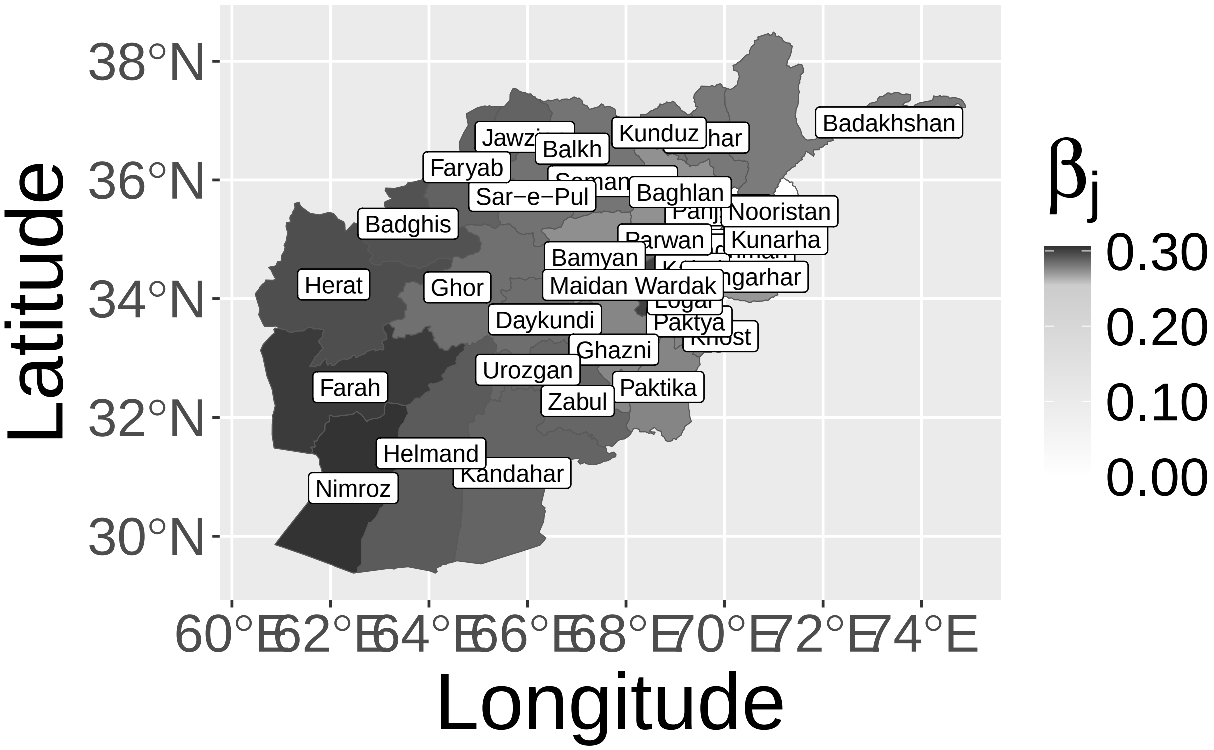
```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.women_age.csv",sep=""))
```

```
# Coefficients of GAMMA_1 (women's age) + Theta at each province in urban area
library(ggplot2)
library(sf)
```

```
map <- read_sf("data_src/map.json")
beta_women_age <- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_women_age <- append(beta_women_age, 0)
    }else{
        beta_women_age <- append(beta_women_age, df$F[which(df$province==n & df$area=="Urban")])
    }
}
map$beta_women_age<- beta_women_age
ggplot(map) +
    geom_sf(aes(fill = beta_women_age)) +
    scale_fill_gradientn(colors = c("white", "gray80", "gray20"),
    values = scales::rescale(c(0, min(beta_women_age[beta_women_age > 0]), max(beta_women_age))),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

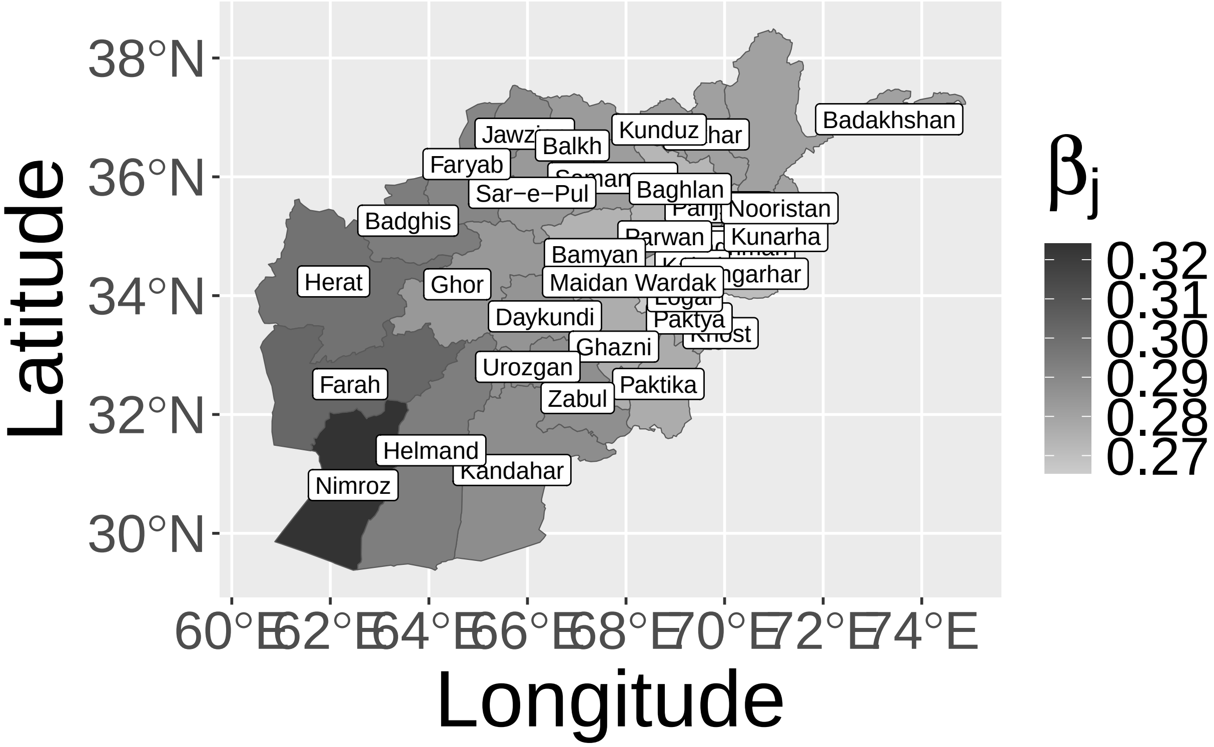


```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_age.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_1 (women's age) + Theta at each province in rural area
library(ggplot2)
library(sf)

map <- read_sf("data_src/map.json")
beta_women_age <- NULL
for (n in map$name){
    beta_women_age <- append(beta_women_age, df$F[which(df$province==n & df$area=="Rural")])
}
map$beta_women_age<- beta_women_age
ggplot(map) +
    geom_sf(aes(fill = beta_women_age)) +
    scale_fill_gradientn(colors = c("gray80", "gray20"),
    values = scales::rescale(c(min(beta_women_age[beta_women_age > 0]), max(beta_women_age))),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),      # Legend title size
  legend.text = element_text(size = 20),        # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```



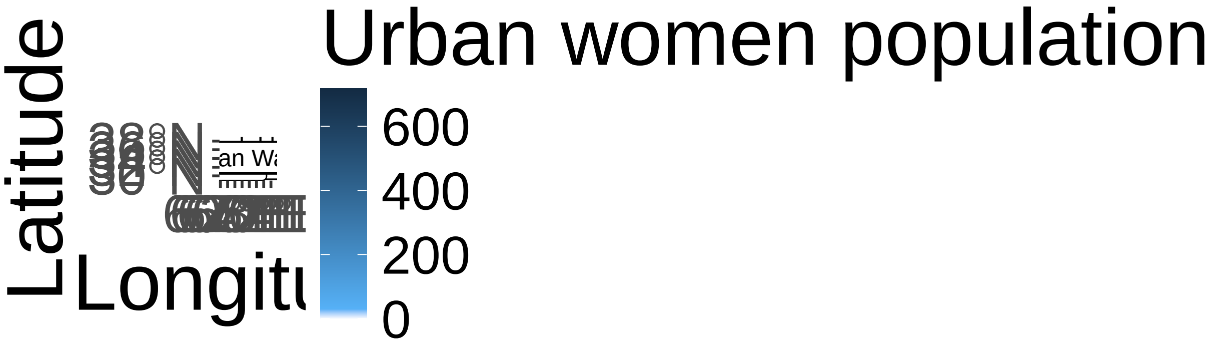
```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_age.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

Plot the Sample size in each stratum

```
map <- read_sf("data_src/map.json")
sample_size <- NULL

for (n in map$name){

    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        sample_size <- append(sample_size, 0)
    }else{
        sample_size <- append(sample_size, df$stratum_sample_size[which(df$province==n & df$area=="Urba
    }
}
map$sample_size <- sample_size
ggplot(map) +
    geom_sf(aes(fill = sample_size)) +
    scale_fill_gradientn(colors = c("white","#56B1F7", "#132B43"),
    values = scales::rescale(c(0,min(sample_size[sample_size>0]), max(sample_size))),
    guide = "colorbar", labels = function(x) sprintf("%d", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill="Urban women population", x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results f
## data
```



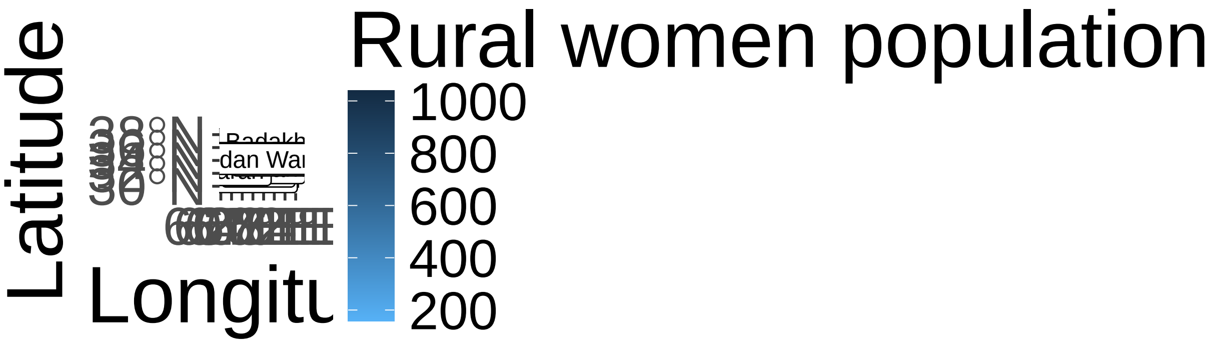
```
dev.print(pdf, paste(IMAGEOUT, "sample_size.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
map <- read_sf("data_src/map.json")
sample_size <- NULL
for (n in map$name){
    sample_size <- append(sample_size, df$stratum_sample_size[which(df$province==n & df$area=="Rural")]
}
```

```
map$sample_size <- sample_size
ggplot(map) +
    geom_sf(aes(fill = sample_size)) +
    scale_fill_gradientn(colors = c("#56B1F7", "#132B43"),
    values = scales::rescale(c(min(sample_size), max(sample_size))),
    guide = "colorbar", labels = function(x) sprintf("%d", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill="Rural women population", x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),      # Legend title size
  legend.text = element_text(size = 20),        # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```



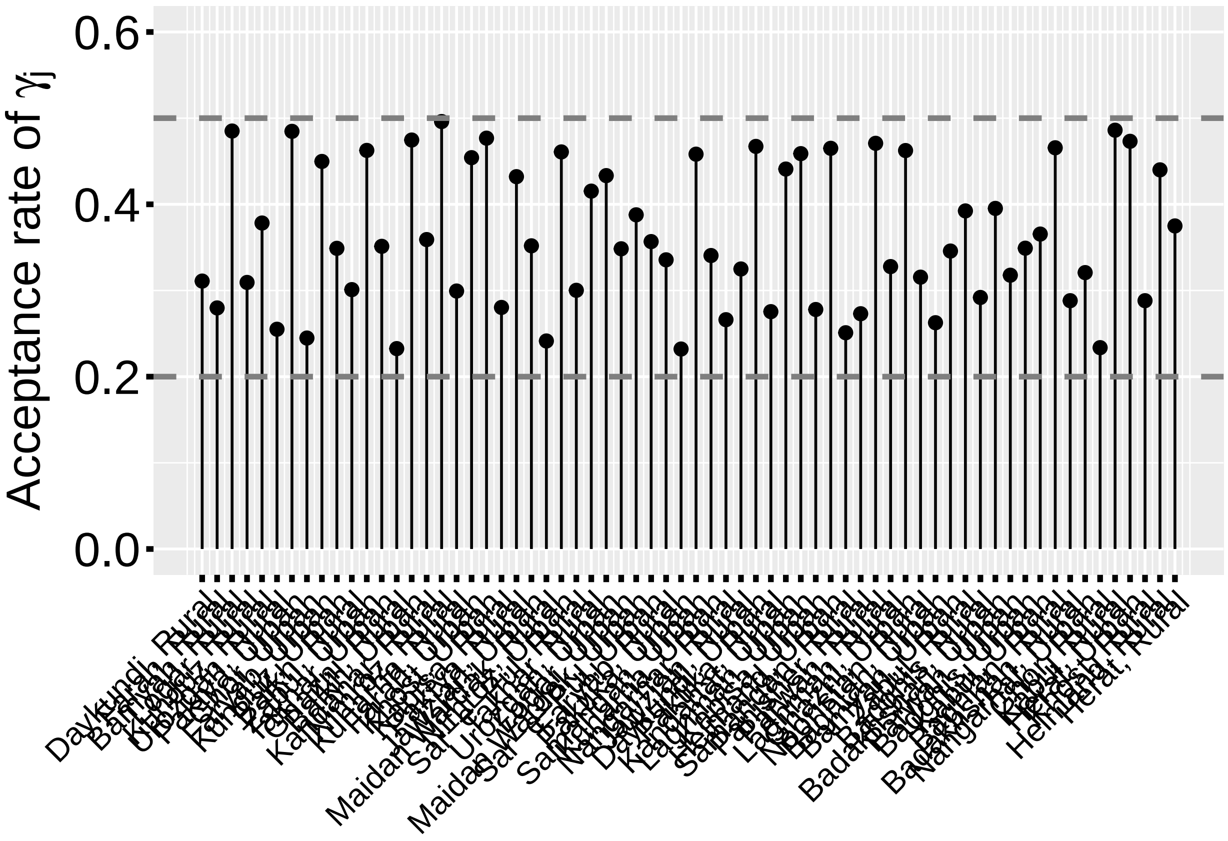
```
dev.print(pdf, paste(IMAGEOUT, "sample_size.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

The acceptance rate of each gamma

```
library("latex2exp")
df <- data.frame(x = 1:66,
                 L = rep(0,66),
                 U = ACCEPT.count/S)
abc <- ggplot(df, aes(x = x, y = U)) +
  labs(y =
       TeX("Acceptance rate of $\\gamma_j$"),
       x = NULL) +
  geom_point(size = 2) +
  geom_linerange(aes(ymax = U, ymin = L)) +
    geom_hline(yintercept = 0.2, color = "gray50", linetype = "dashed", size = 1) +
  geom_hline(yintercept = 0.5, color = "gray50", linetype = "dashed", size = 1) +
  scale_y_continuous(limits = c(0, 0.6)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
```

```r
print(abc)
```

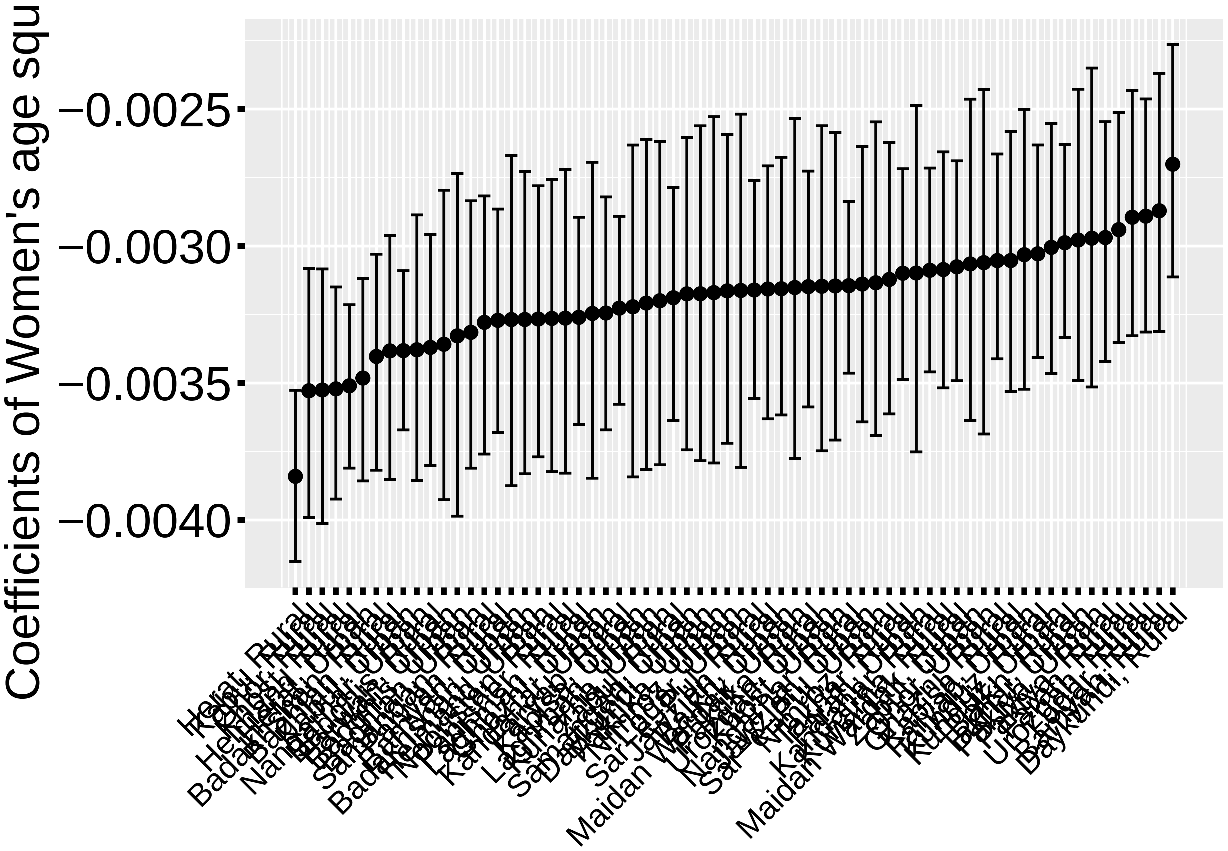


```r
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
dev.print(pdf, paste(IMAGEOUT, "ACCEPT.rate.GAMMA.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of women_age2

```r
i <- 1
GAMMA_women_age2 <- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 3])
    }
    GAMMA_women_age2 <- rbind(GAMMA_women_age2, gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_women_age2+rep(THETA.post.mat[3,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
           province = stratum_province,
               area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of Women's age squared", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



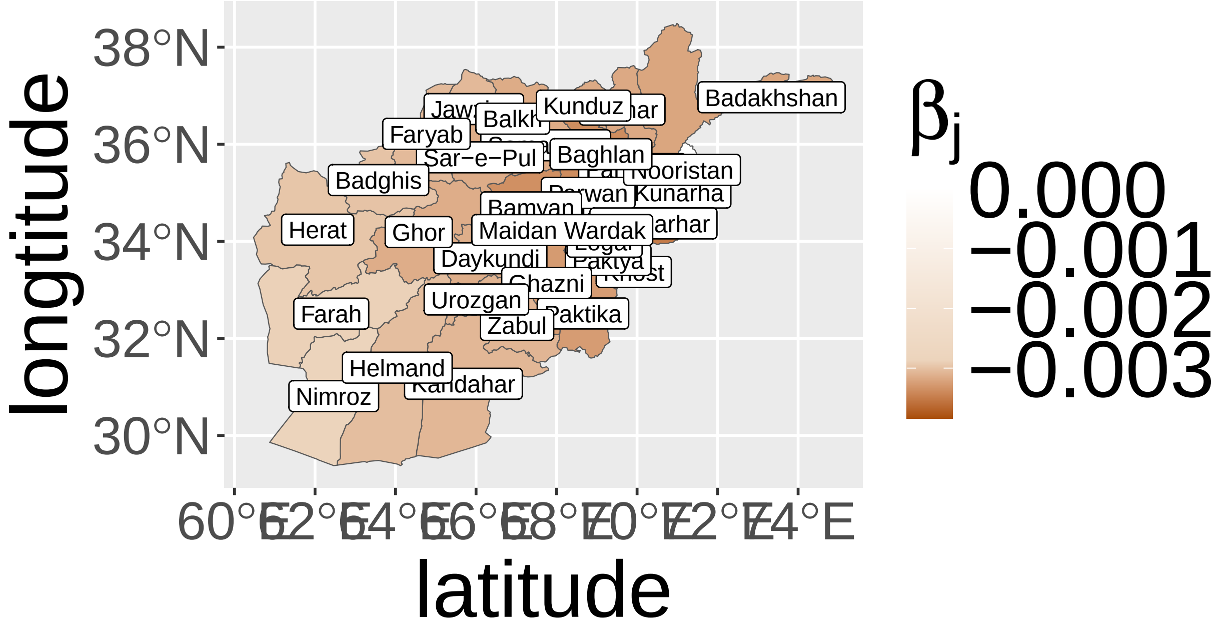
```
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
dev.print(pdf, paste(IMAGEOUT, "BETA.women_age2.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.women_age2.csv",sep=""))
```

```
# Coefficients of GAMMA_2 (women's age squared) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_women_age2 <- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_women_age2 <- append(beta_women_age2, 0)
    }else{
        beta_women_age2 <- append(beta_women_age2, df$F[which(df$province==n & df$area=="Urban")])
    }
}
map$beta_women_age2<- beta_women_age2
ggplot(map) +
    geom_sf(aes(fill = beta_women_age2)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white"),
    values = scales::rescale(c(min(beta_women_age2), max(beta_women_age2[beta_women_age2 < 0]), 0)),
    guide = "colorbar", labels = function(x) sprintf("%.3f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 30),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
    )
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results f
## data
```



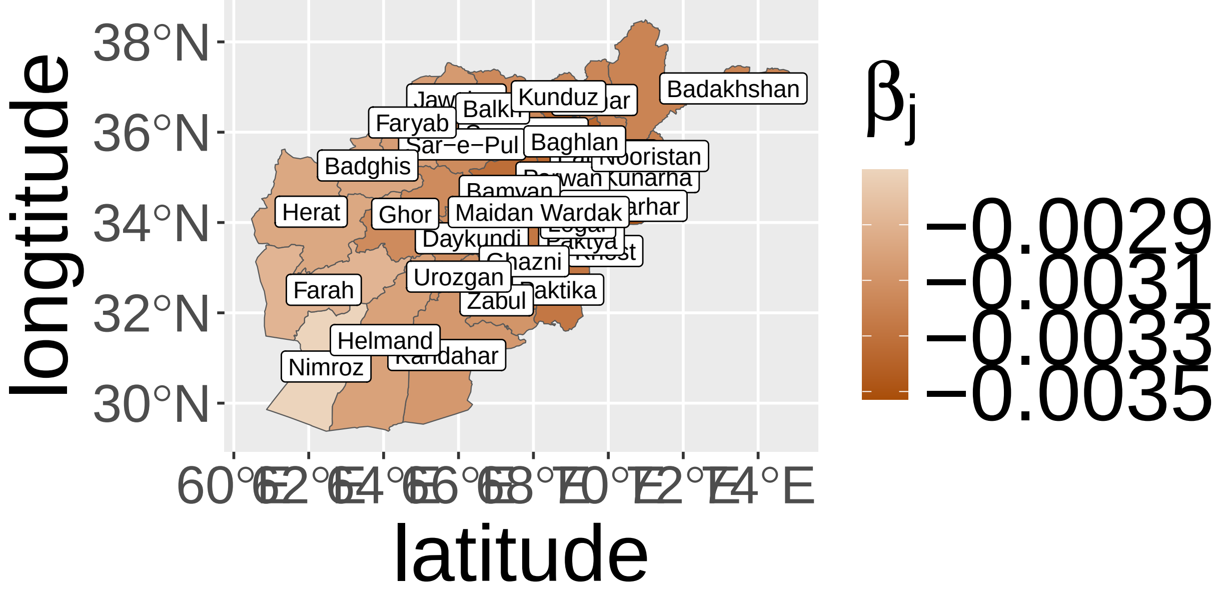
```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_age2.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_2 (women's age squared) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_women_age2 <- NULL
for (n in map$name){
    beta_women_age2 <- append(beta_women_age2, df$F[which(df$province==n & df$area=="Rural")])
}
```

```
map$beta_women_age2<- beta_women_age2
ggplot(map) +
    geom_sf(aes(fill = beta_women_age2)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc"),
    values = scales::rescale(c(min(beta_women_age2), max(beta_women_age2[beta_women_age2 < 0]))),
    guide = "colorbar", labels = function(x) sprintf("%.4f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 30),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
    )
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results f
## data
```



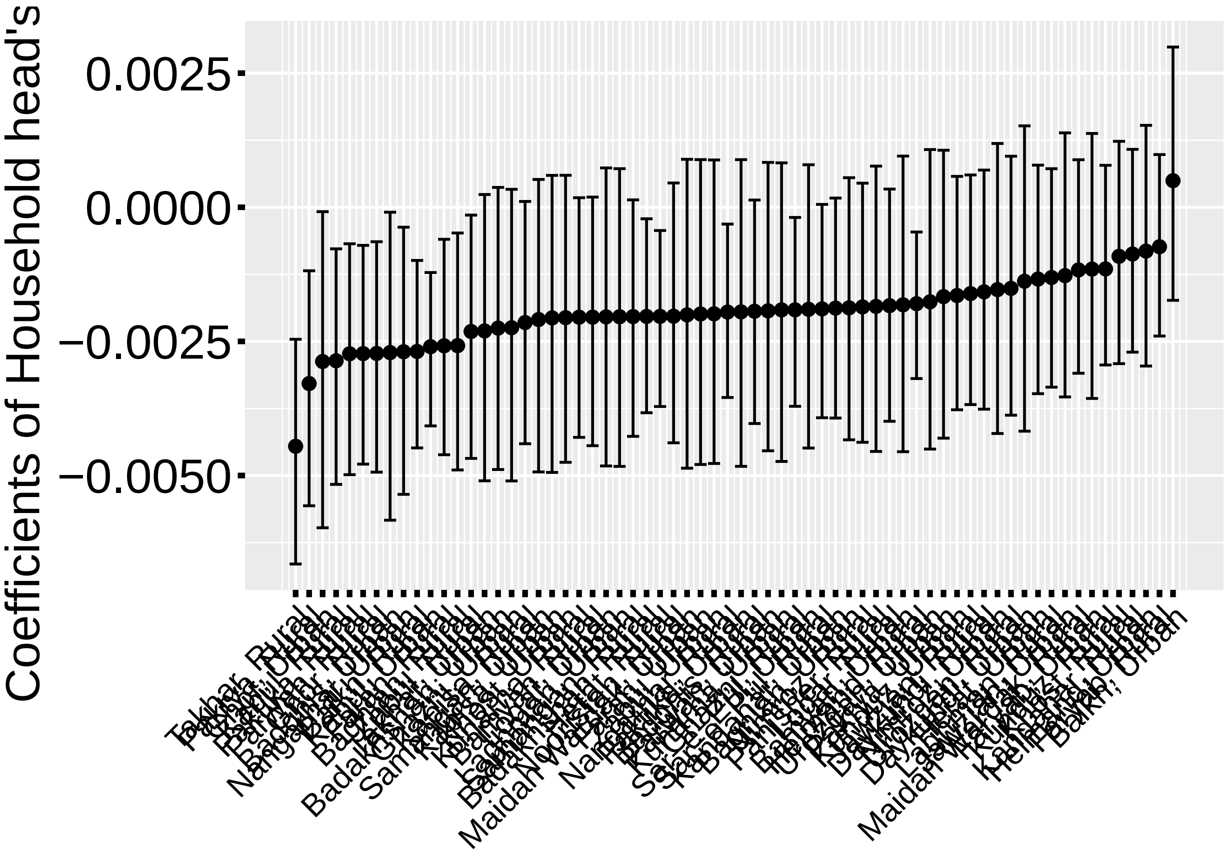
```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_age2.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of HH_age

```
i <- 1
GAMMA_HH_age<- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 4])
    }
    GAMMA_HH_age<- rbind(GAMMA_HH_age, gamma_ij)
    i <- i+1
}
```

```
qE<-apply(GAMMA_HH_age+rep(THETA.post.mat[4,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of Household head's age", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



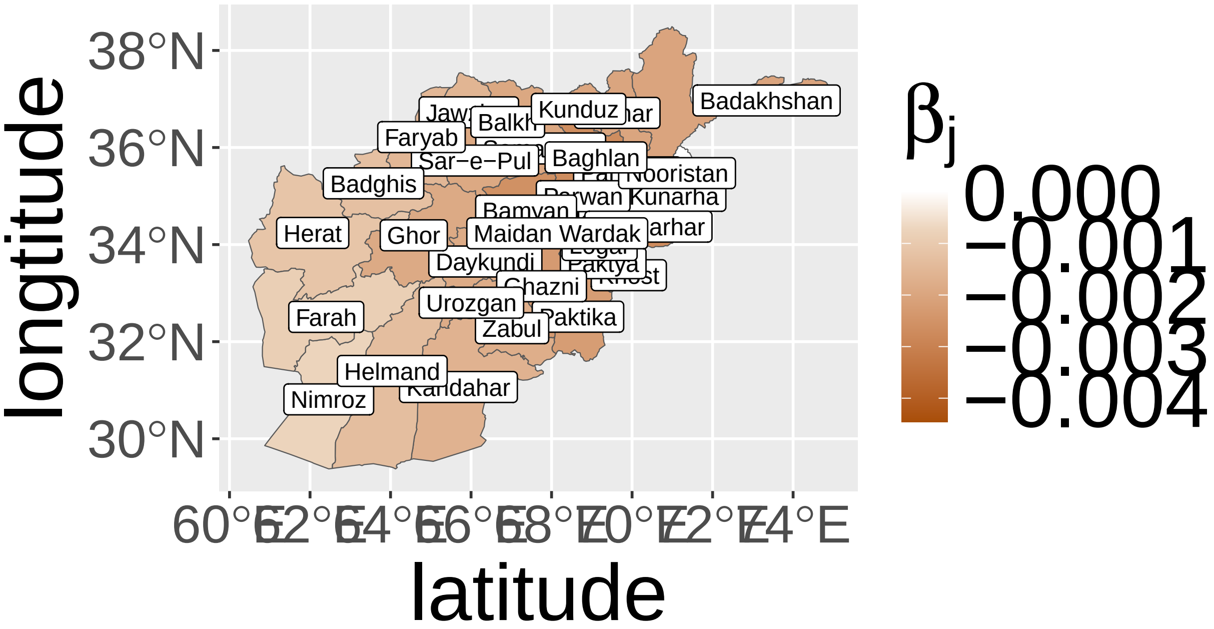
```
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
dev.print(pdf, paste(IMAGEOUT, "BETA.HH_age.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

```r
write.csv(df, paste(IMAGEOUT,"qE.BETA.HH_age.csv",sep=""))
```

```r
# Coefficients of GAMMA_3 (Household head's age) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_HH_age<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_HH_age<- append(beta_HH_age, 0)
    }else{
        #if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area==
        #{
        #   significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
        #}else{
        #   significant_gamma <- 0
        #}
        #beta_HH_age<- append(beta_HH_age, significant_gamma)
        beta_HH_age<- append(beta_HH_age, df$F[which(df$province==n & df$area=="Urban")])
    }
}
map$beta_HH_age<- beta_HH_age
ggplot(map) +
    geom_sf(aes(fill = beta_HH_age)) +
#   scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
#   values = scales::rescale(c(min(beta_HH_age), max(beta_HH_age[beta_HH_age < 0]),0, min(beta_HH_age[b
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white"),
    values = scales::rescale(c(min(beta_HH_age), max(beta_HH_age[beta_HH_age < 0]),0)),
    guide = "colorbar", labels = function(x) sprintf("%.3f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 30),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
    )
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results f
## data
```

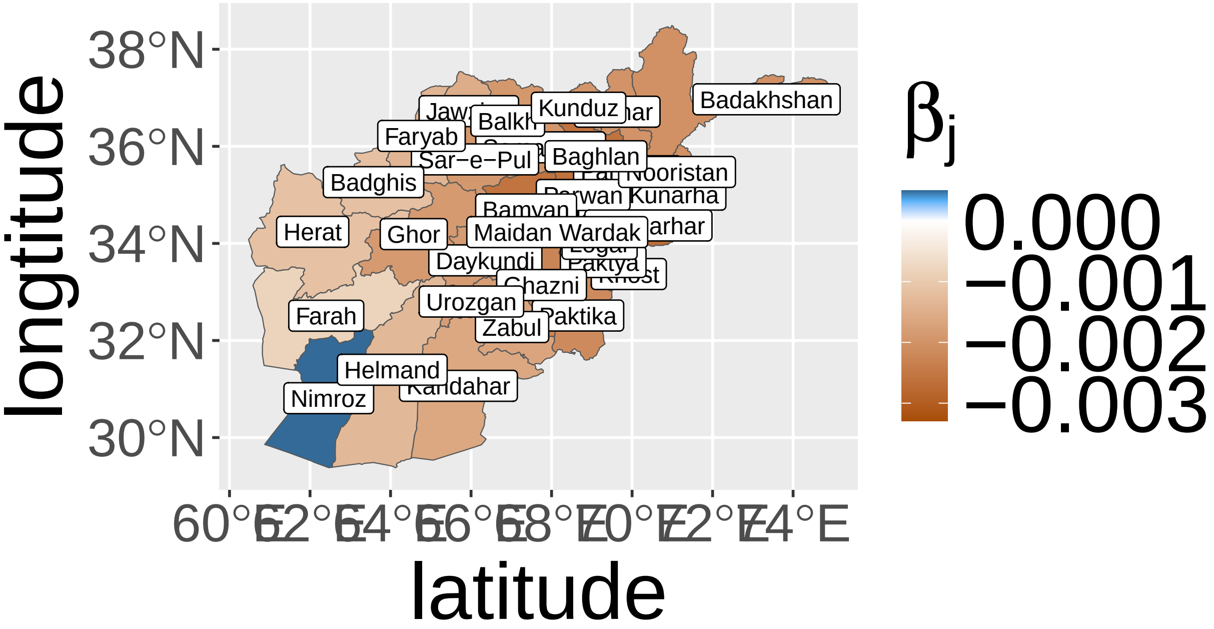


```
dev.print(pdf, paste(IMAGEOUT, "BETA.HH_age.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_3 (Household head's age) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_HH_age<- NULL
for (n in map$name){
    #if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rur
    #{
    #   significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    #}else{
    #   significant_gamma <- 0
    #}
    #beta_HH_age<- append(beta_HH_age, significant_gamma)
    beta_HH_age<- append(beta_HH_age, df$F[which(df$province==n & df$area=="Rural")])
}
map$beta_HH_age<- beta_HH_age
ggplot(map) +
    geom_sf(aes(fill = beta_HH_age)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_HH_age), max(beta_HH_age[beta_HH_age<0]), 0, min(beta_HH_age[be
    guide = "colorbar", labels = function(x) sprintf("%.3f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 30),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
    )
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm): collapsing to unique 'x' valu
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm): collapsing to unique 'x' valu
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm): collapsing to unique 'x' valu
```

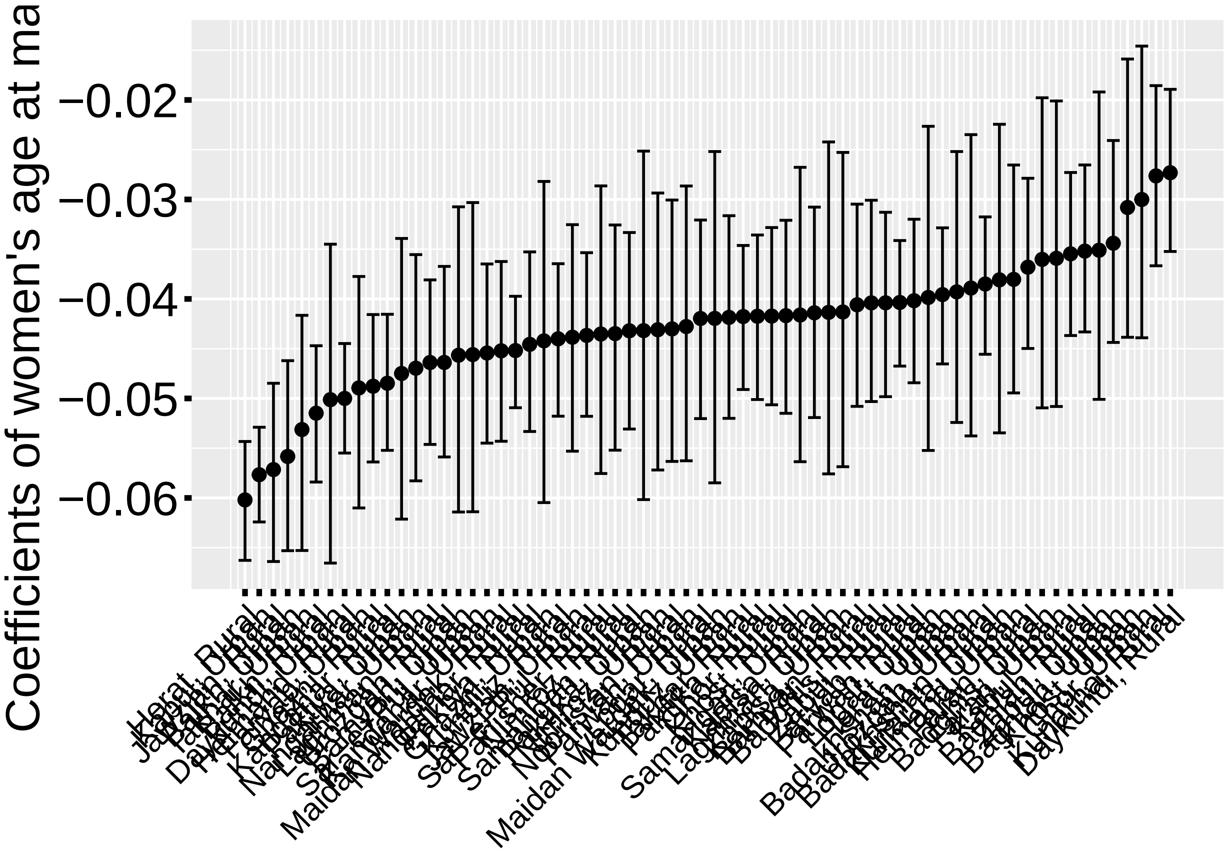


```
dev.print(pdf, paste(IMAGEOUT, "BETA.HH_age.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of women_agem

```
i <- 1
GAMMA_women_agem <- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 5])
    }
    GAMMA_women_agem <- rbind(GAMMA_women_agem, gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_women_agem+rep(THETA.post.mat[5,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of women's age at marriage", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



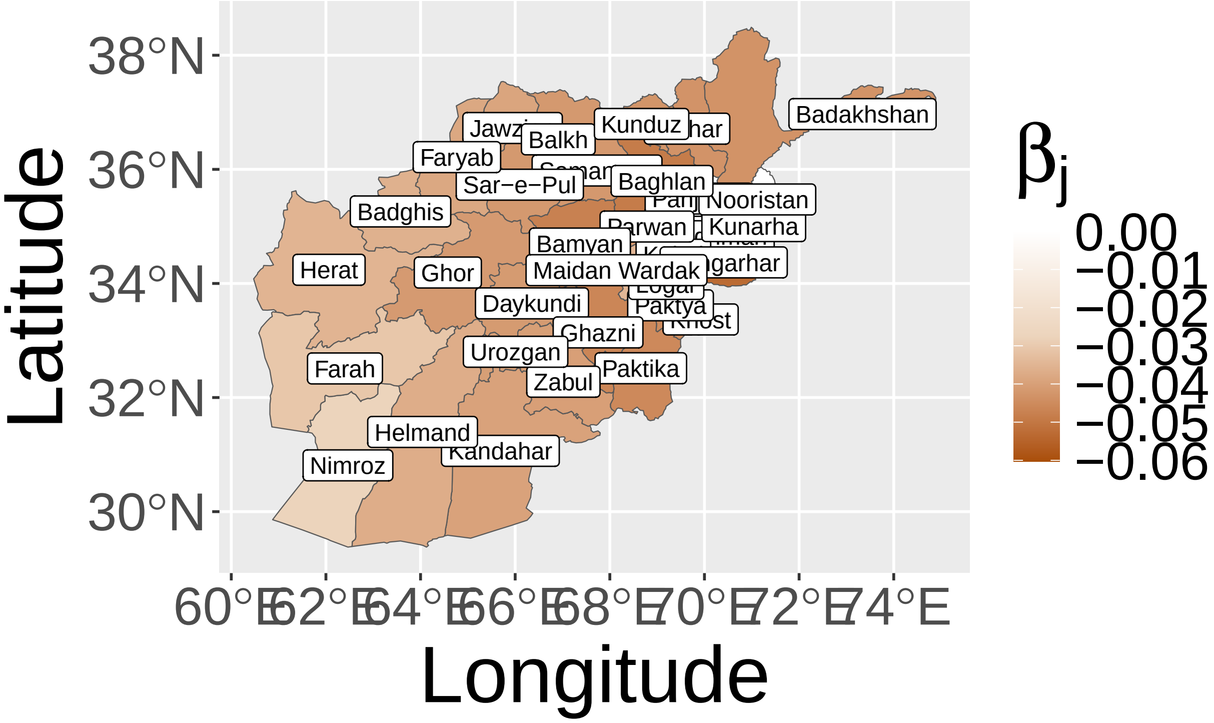
```
#dev.print(device = postscript, paste(IMAGEOUT, "E_province.eps",sep=""),width=8,height=8, horizontal =
dev.print(pdf, paste(IMAGEOUT, "BETA.women_agem.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.women_agem.csv",sep=""))
```

```
# Coefficients of GAMMA_5 (women's age at marriage) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_women_agem<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_women_agem<- append(beta_women_agem, 0)
    }else{
        if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area=="
        {
            significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
        }else{
            significant_gamma <- 0
        }
        beta_women_agem<- append(beta_women_agem, significant_gamma)
    }
}
map$beta_women_agem<- beta_women_agem
ggplot(map) +
    geom_sf(aes(fill = beta_women_agem)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white"),
    values = scales::rescale(c(min(beta_women_agem), max(beta_women_agem[beta_women_agem< 0]), 0)),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),      # Legend title size
  legend.text = element_text(size = 20),        # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results f
## data
```



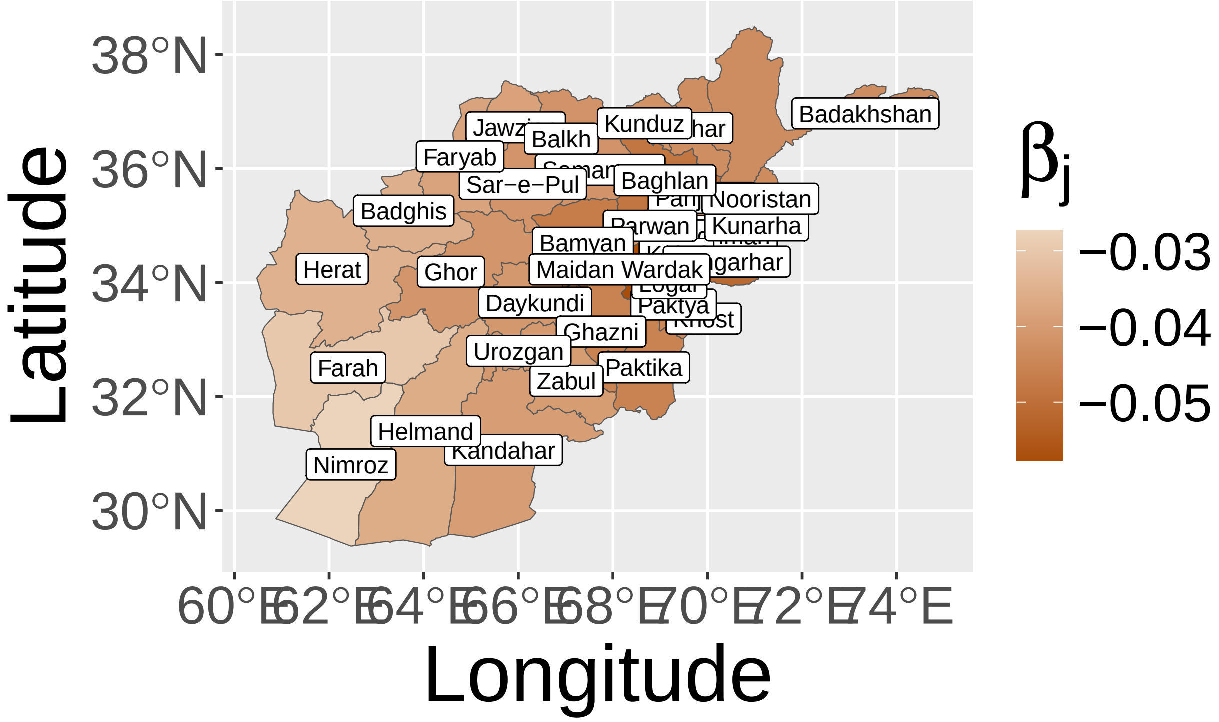
```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_agem.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_5 (women's age at marriage) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_women_agem<- NULL
```

```
for (n in map$name){
    if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rural
    {
        significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    }else{
        significant_gamma <- 0
    }
    beta_women_agem<- append(beta_women_agem, significant_gamma)
}
map$beta_women_agem<- beta_women_agem
ggplot(map) +
    geom_sf(aes(fill = beta_women_agem)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc"),
    values = scales::rescale(c(min(beta_women_agem), max(beta_women_agem))),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

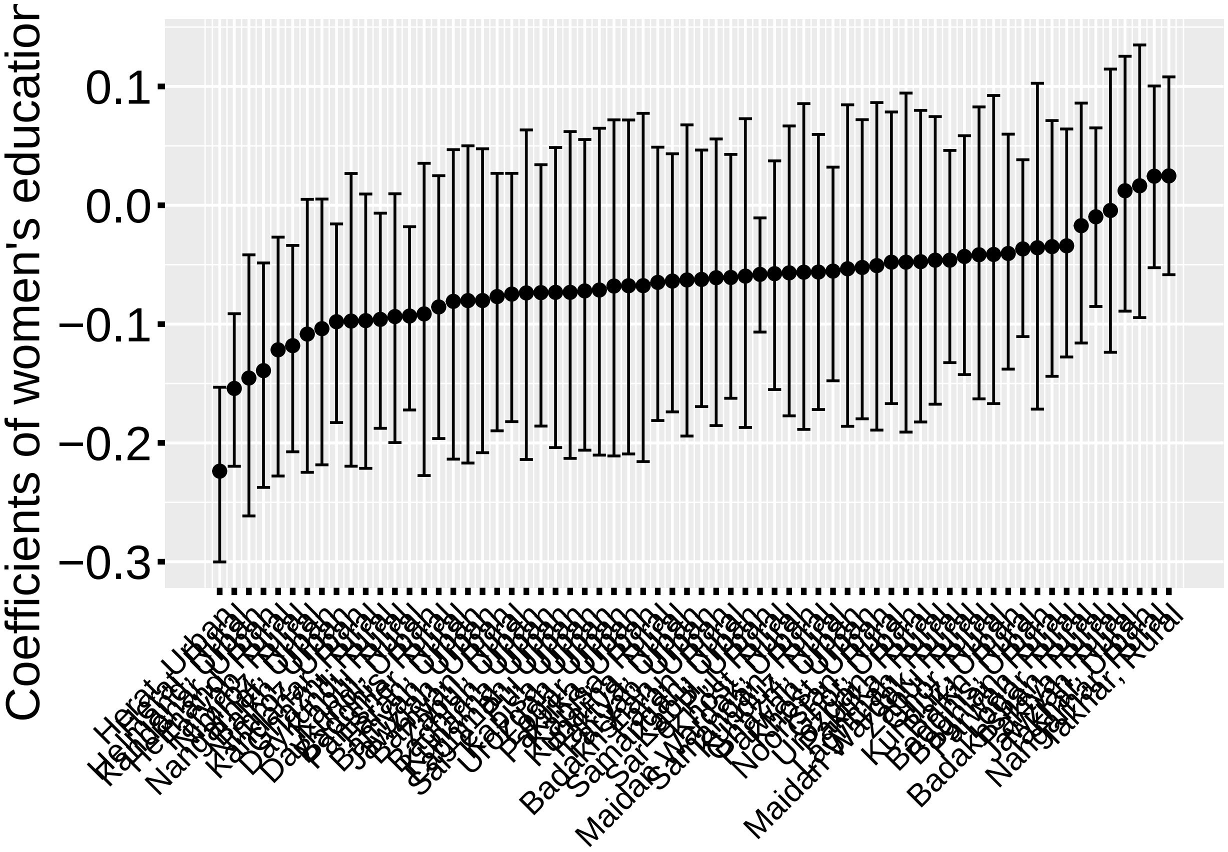


```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_agem.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of women_edu

```
i <- 1
GAMMA_women_edu <- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 6])
    }
    GAMMA_women_edu <- rbind(GAMMA_women_edu , gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_women_edu+rep(THETA.post.mat[6,],66) ,1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of women's education level", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



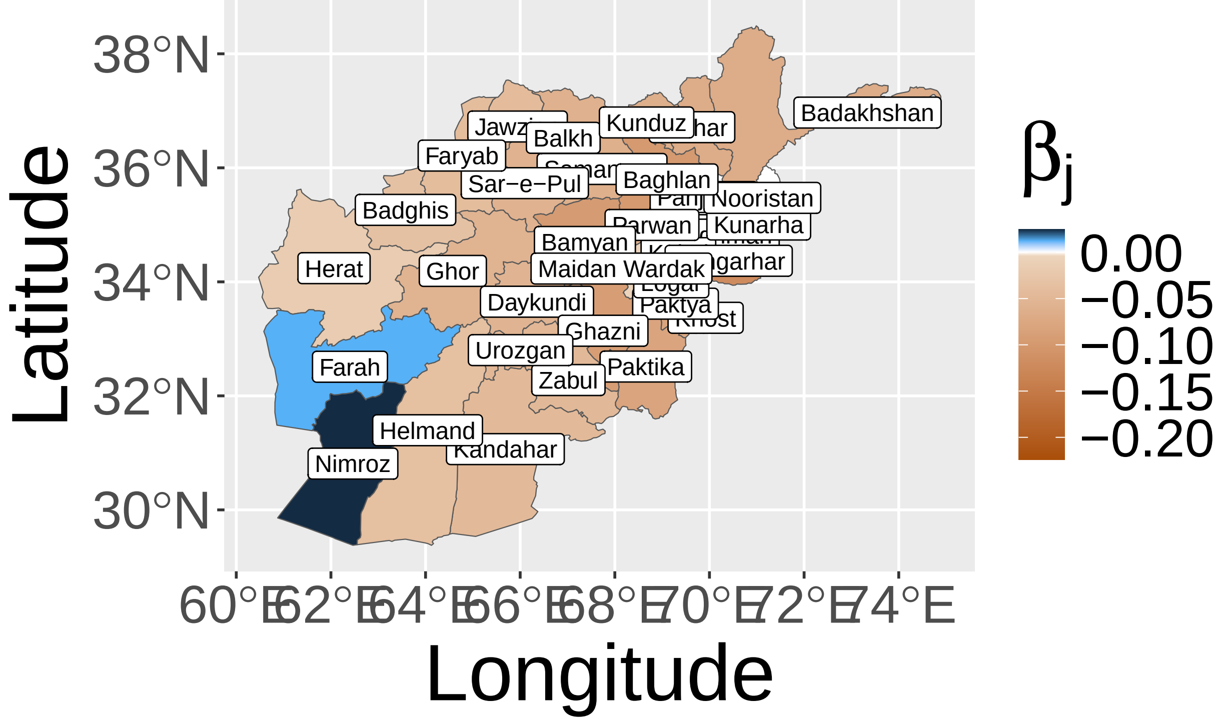
```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_edu.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.women_edu.csv",sep=""))
```

```
# Coefficients of GAMMA_6 (women's education) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_women_edu<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_women_edu<- append(beta_women_edu, 0)
    }else{
        #if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area==
        #{
        #   significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
        #}else{
        #   significant_gamma <- 0
        #}
        #beta_women_edu<- append(beta_women_edu, significant_gamma)
        beta_women_edu<- append(beta_women_edu, df$F[which(df$province==n & df$area=="Urban")])
    }
}
map$beta_women_edu<- beta_women_edu
ggplot(map) +
    geom_sf(aes(fill = beta_women_edu)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_women_edu), max(beta_women_edu[beta_women_edu< 0]),0, min(beta_w
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),      # Legend title size
  legend.text = element_text(size = 20),        # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```



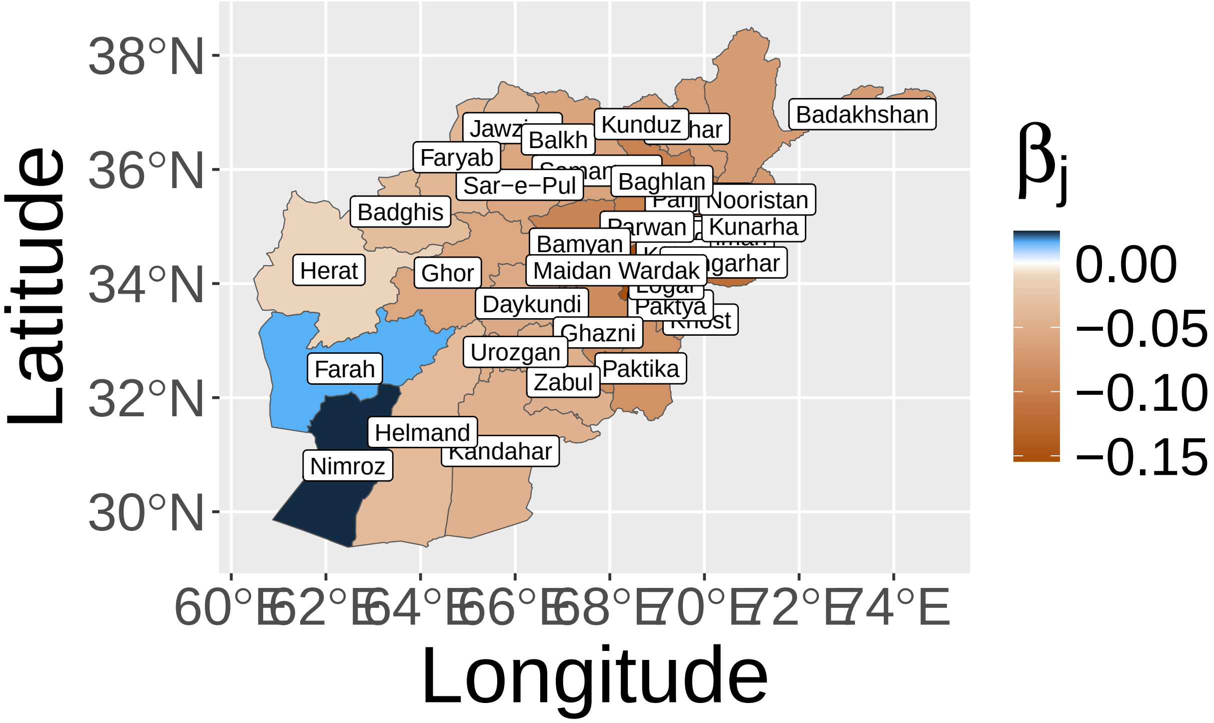
```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_edu.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_6 (women's education) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_women_edu<- NULL
for (n in map$name){
    #if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rur
    #{
    #   significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    #}else{
    #   significant_gamma <- 0
    #}
    #beta_women_edu<- append(beta_women_edu, significant_gamma)
    beta_women_edu<- append(beta_women_edu, df$F[which(df$province==n & df$area=="Rural")])
```

```
}
map$beta_women_edu<- beta_women_edu
ggplot(map) +
    geom_sf(aes(fill = beta_women_edu)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_women_edu), max(beta_women_edu[beta_women_edu< 0]),0, min(beta_
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="Longitude", y="Latitude") +
    theme(
  legend.title = element_text(size = 30),      # Legend title size
  legend.text = element_text(size = 20),        # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

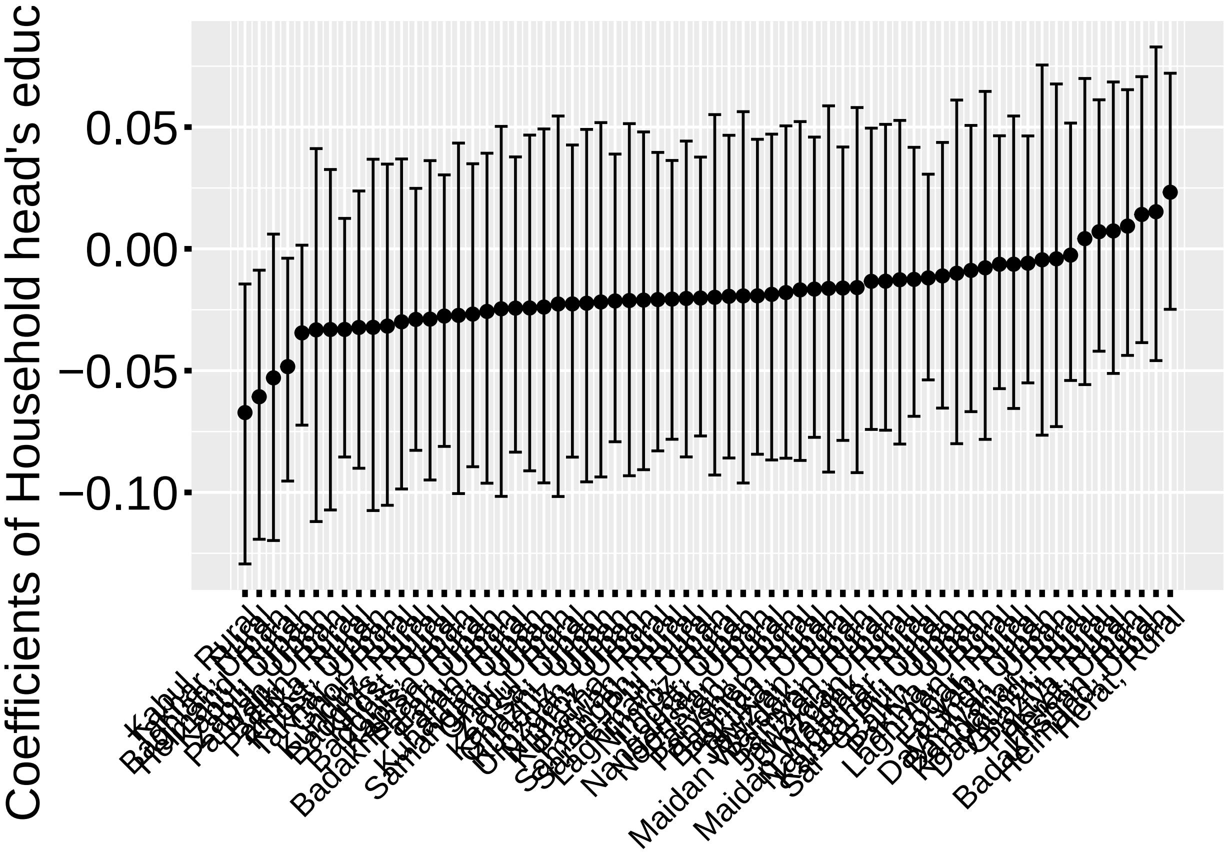


```
dev.print(pdf, paste(IMAGEOUT, "BETA.women_edu.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of HH_edu

```
i <- 1
GAMMA_HH_edu <- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 7])
    }
    GAMMA_HH_edu <- rbind(GAMMA_HH_edu , gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_HH_edu +rep(THETA.post.mat[7,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of Household head's education level", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



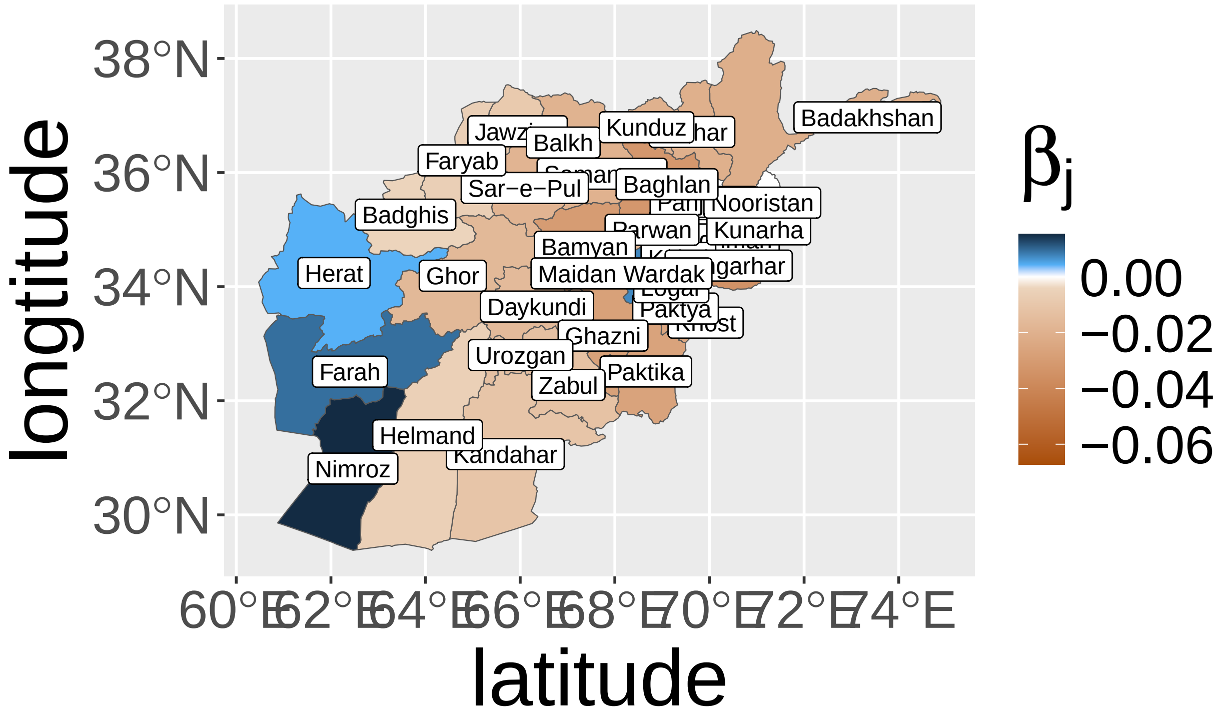
```
dev.print(pdf, paste(IMAGEOUT, "BETA.HH_edu.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.HH_edu.csv",sep=""))
```

```
# Coefficients of GAMMA_7 (Household head's education) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_HH_edu<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_HH_edu<- append(beta_HH_edu, 0)
    }else{
        #if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area==
        #{
        #   significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
        #}else{
        #   significant_gamma <- 0
        #}
        #beta_HH_edu<- append(beta_HH_edu, significant_gamma)
        beta_HH_edu<- append(beta_HH_edu, df$F[which(df$province==n & df$area=="Urban")])
    }
}
map$beta_HH_edu<- beta_HH_edu
ggplot(map) +
    geom_sf(aes(fill = beta_HH_edu)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_HH_edu), max(beta_HH_edu[beta_HH_edu< 0]),0, min(beta_HH_edu[be
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

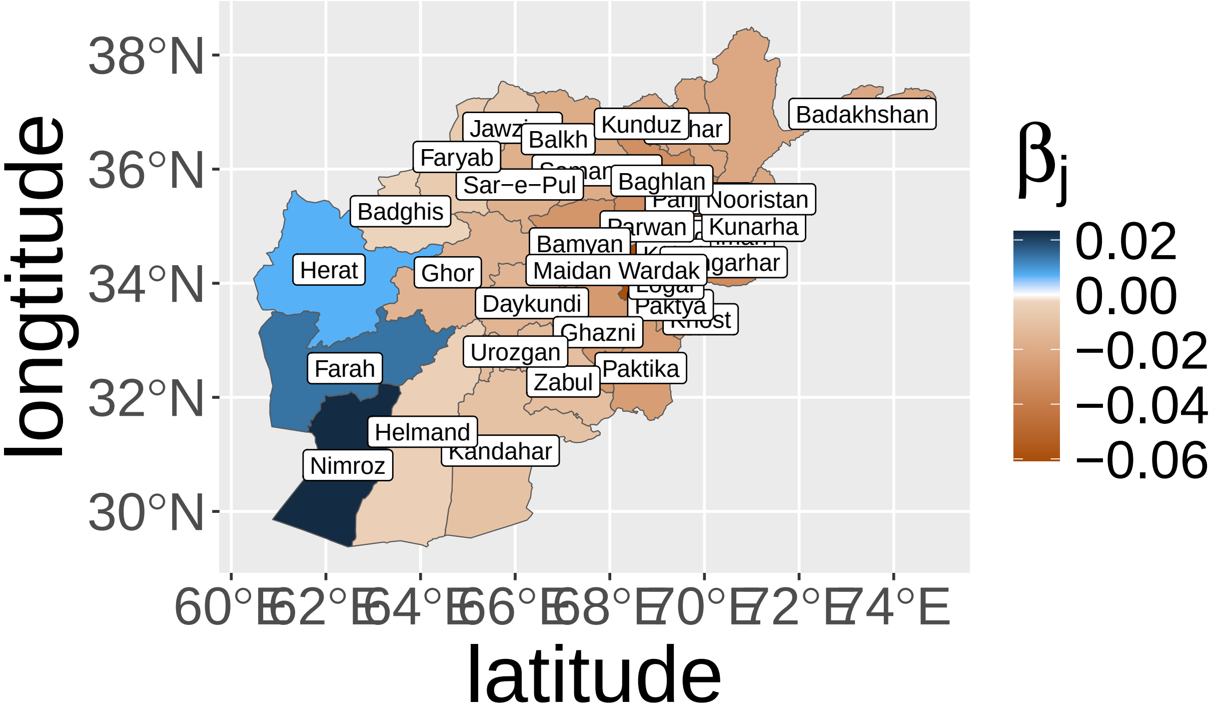


```r
dev.print(pdf, paste(IMAGEOUT, "BETA.HH_edu.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```r
# Coefficients of GAMMA_7 (Household head's education) at each province in rural area
map <- read_sf("data_src/map.json")
beta_HH_edu<- NULL
for (n in map$name){
    #if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rur
    #{
    #   significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    #}else{
    #   significant_gamma <- 0
    #}
    #beta_HH_edu<- append(beta_HH_edu, significant_gamma)
    beta_HH_edu<- append(beta_HH_edu, df$F[which(df$province==n & df$area=="Rural")])
}
map$beta_HH_edu<- beta_HH_edu
ggplot(map) +
    geom_sf(aes(fill = beta_HH_edu)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_HH_edu), max(beta_HH_edu[beta_HH_edu< 0]),0, min(beta_HH_edu[be
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),      # Legend title size
  legend.text = element_text(size = 20),        # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

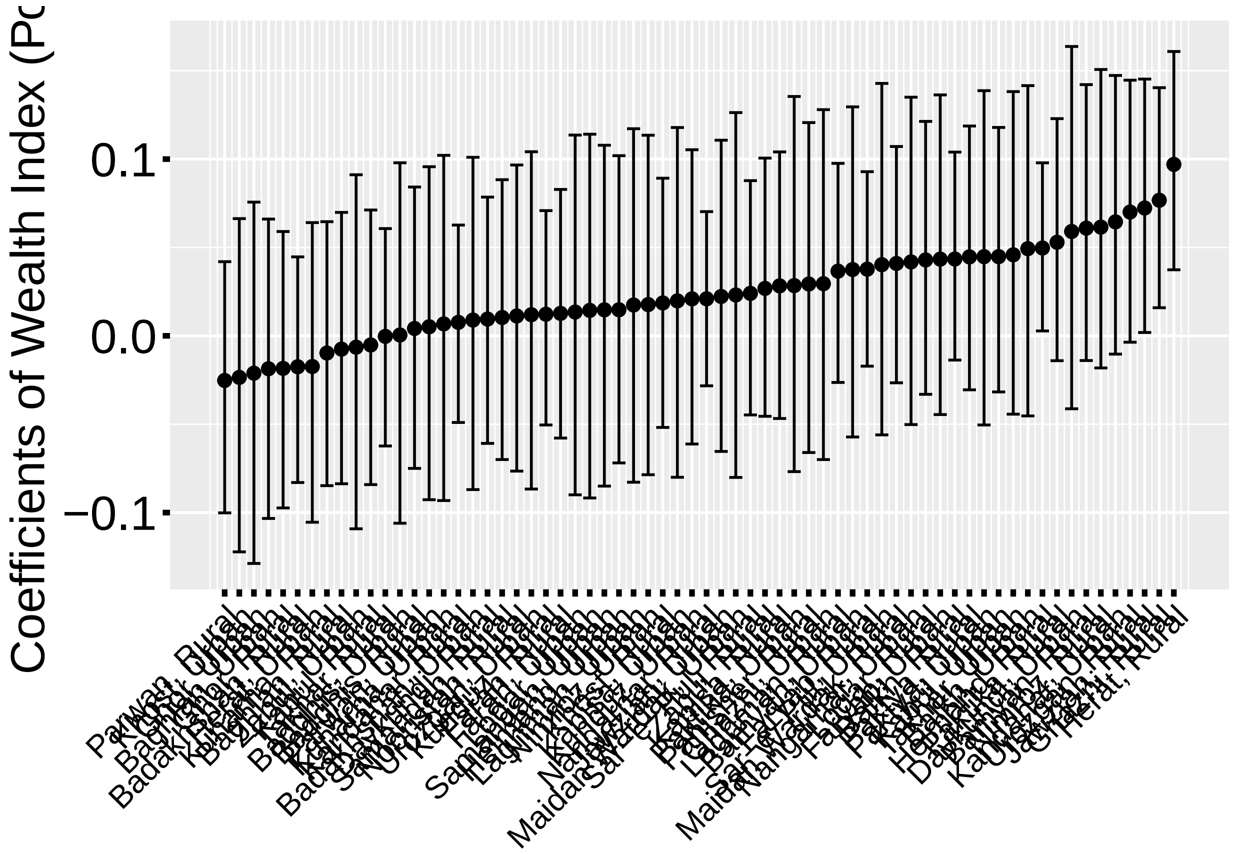


```r
dev.print(pdf, paste(IMAGEOUT, "BETA.HH_edu.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of wealth_index_Poor

```r
i <- 1
GAMMA_poor<- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 8])
    }
    GAMMA_poor <- rbind(GAMMA_poor , gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_poor+rep(THETA.post.mat[8,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
            province = stratum_province,
                area = stratum_area,
stratum_sample_size= stratum_sample_size,
                   F = qE[2,][index],
                   L = qE[1,][index],
                   U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of Wealth Index (Poor)", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
dev.print(pdf, paste(IMAGEOUT, "BETA.windex.poor.pdf",sep=""),width=16,height=8)
```

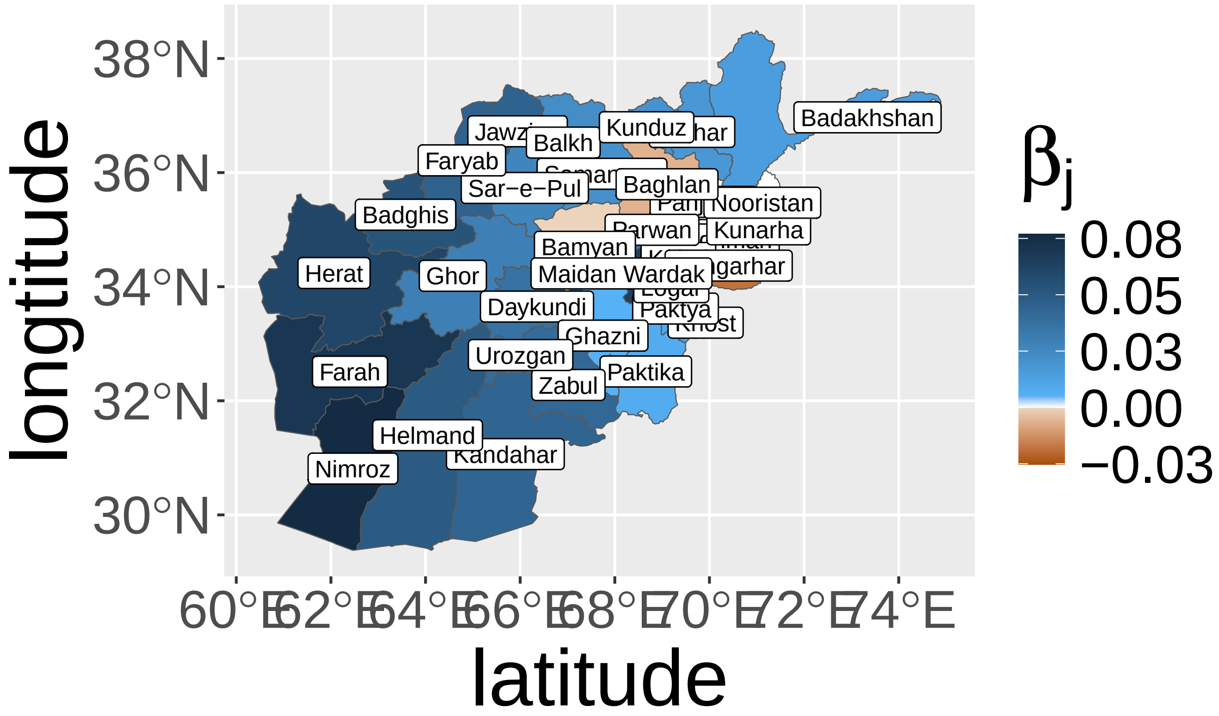
```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.windex.poor.csv",sep=""))
```

```
# Coefficients of GAMMA_8 (Wealth - poor) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_windex_poor<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_windex_poor<- append(beta_windex_poor, 0)
    }else{
        #if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area==
        #{
        #   significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
```

```r
        #}else{
        #   significant_gamma <- 0
        #}
        #beta_windex_poor<- append(beta_windex_poor, significant_gamma)
        beta_windex_poor<- append(beta_windex_poor, df$F[which(df$province==n & df$area=="Urban")])
    }
}
# Add beta_windex_poor to the map data
map$beta_windex_poor<- beta_windex_poor

# Define the custom color scale
color_scale <- c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43")

# Define the values for the color scale
min_value <- min(beta_windex_poor, na.rm = TRUE)
max_negative_value <- max(beta_windex_poor[beta_windex_poor < 0], na.rm = TRUE)
min_positive_value <- min(beta_windex_poor[beta_windex_poor > 0], na.rm = TRUE)
max_value <- max(beta_windex_poor, na.rm = TRUE)

ggplot(map) +
    geom_sf(aes(fill = beta_windex_poor)) +
    scale_fill_gradientn(colors = color_scale,
    values = scales::rescale(c(min_value, max_negative_value, 0, min_positive_value, max_value)),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

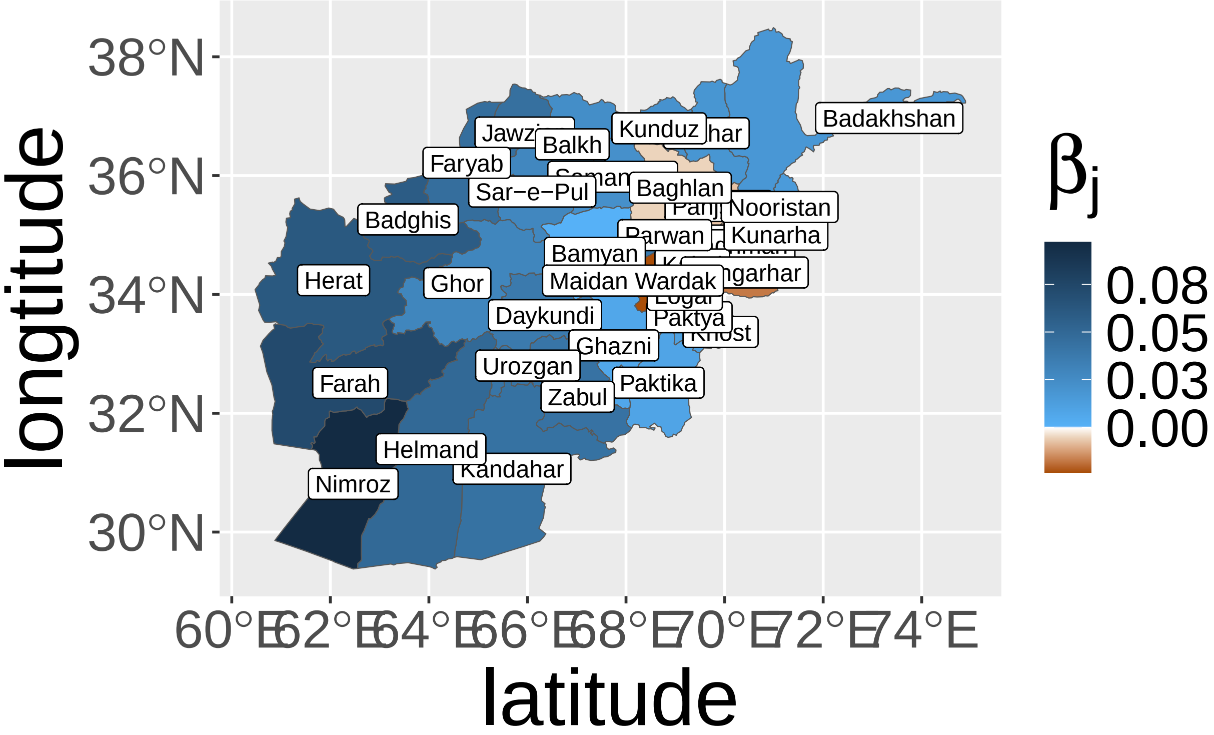


```
dev.print(pdf, paste(IMAGEOUT, "BETA.windex.poor.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_8 (Wealth - poor) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_windex_poor<- NULL
for (n in map$name){
    #if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rur
    #{
    #   significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    #}else{
    #   significant_gamma <- 0
    #}
    #beta_windex_poor<- append(beta_windex_poor, significant_gamma)
    beta_windex_poor<- append(beta_windex_poor, df$F[which(df$province==n & df$area=="Rural")])
}
map$beta_windex_poor<- beta_windex_poor
ggplot(map) +
    geom_sf(aes(fill = beta_windex_poor)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_windex_poor), max(beta_windex_poor[beta_windex_poor<0]), 0, min
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

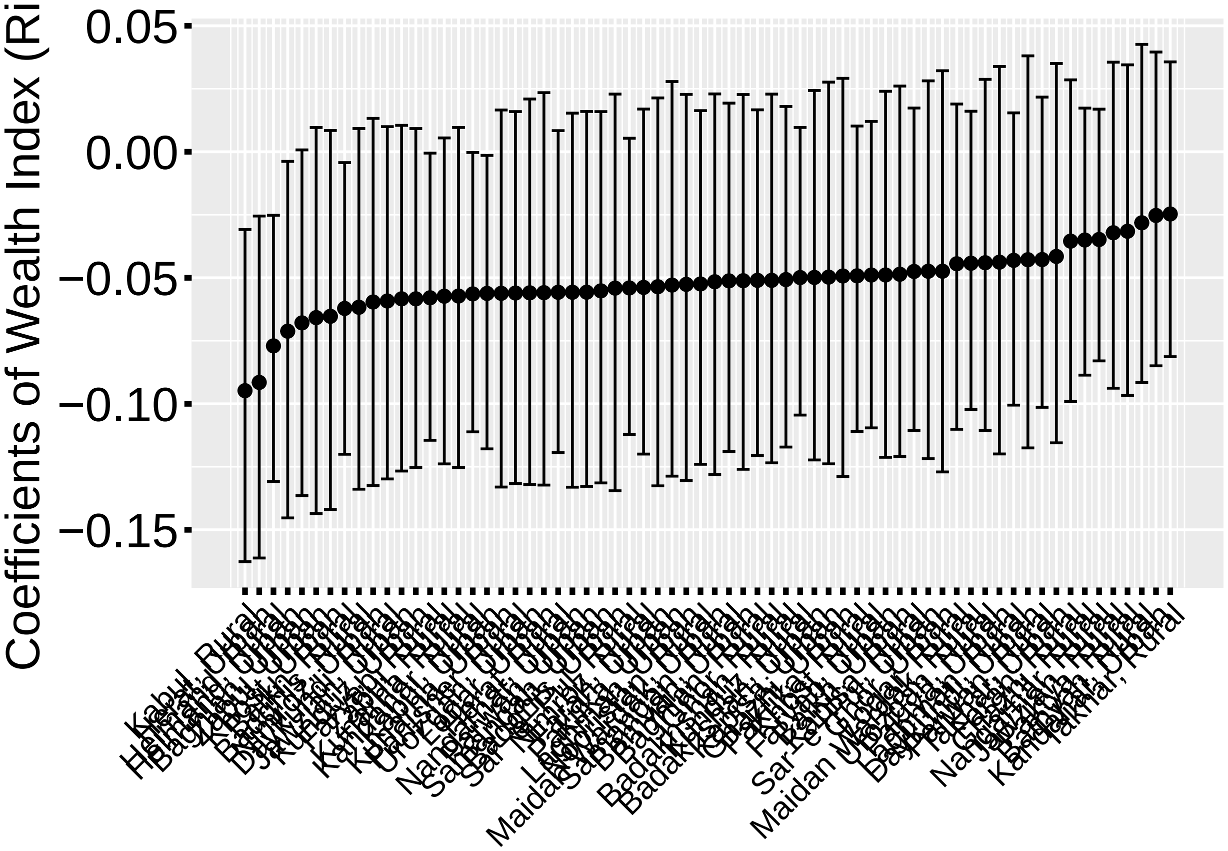


```
dev.print(pdf, paste(IMAGEOUT, "BETA.windex.poor.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of wealth_index_Rich

```
i <- 1
GAMMA_rich<- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 9])
    }
    GAMMA_rich <- rbind(GAMMA_rich , gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_rich+rep(THETA.post.mat[9,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of Wealth Index (Rich)", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
dev.print(pdf, paste(IMAGEOUT, "BETA.windex.rich.pdf",sep=""),width=16,height=8)
```

```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.windex.rich.csv",sep=""))
```

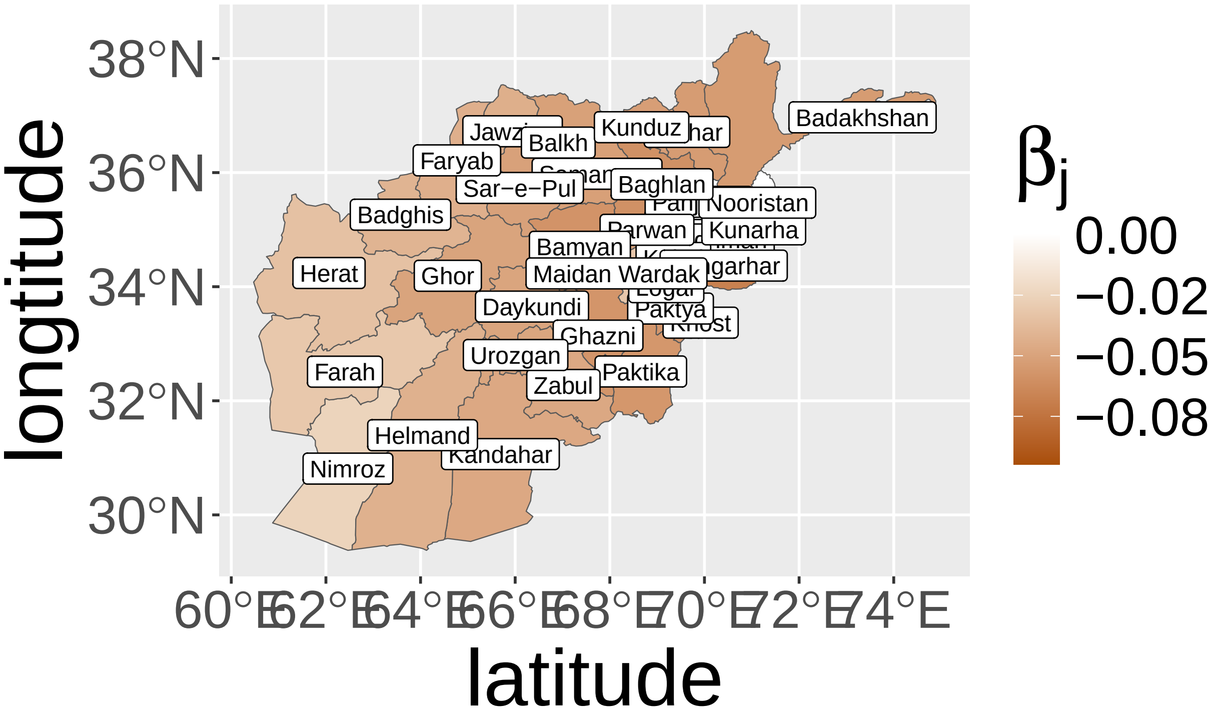
```
# Coefficients of GAMMA_9 (Wealth - rich) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_windex_rich<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_windex_rich<- append(beta_windex_rich, 0)
    }else{
        #if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area==
        #{
```

```
        #   significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
        #}else{
        #   significant_gamma <- 0
        #}
        #beta_windex_rich<- append(beta_windex_rich, significant_gamma)
        beta_windex_rich<- append(beta_windex_rich, df$F[which(df$province==n & df$area=="Urban")])
    }
}
# Add beta_windex_rich to the map data
map$beta_windex_rich<- beta_windex_rich

# Define the custom color scale
color_scale <- c("#a94e08", "#ecd4bc", "white")

# Define the values for the color scale
min_value <- min(beta_windex_rich, na.rm = TRUE)
max_negative_value <- max(beta_windex_rich[beta_windex_rich < 0], na.rm = TRUE)
min_positive_value <- min(beta_windex_rich[beta_windex_rich > 0], na.rm = TRUE)
```

```
## Warning in min(beta_windex_rich[beta_windex_rich > 0], na.rm = TRUE): no non-missing arguments to mi
```

```
max_value <- max(beta_windex_rich, na.rm = TRUE)

ggplot(map) +
    geom_sf(aes(fill = beta_windex_rich)) +
    scale_fill_gradientn(colors = color_scale,
    values = scales::rescale(c(min_value, max_negative_value, 0)),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results f
## data
```

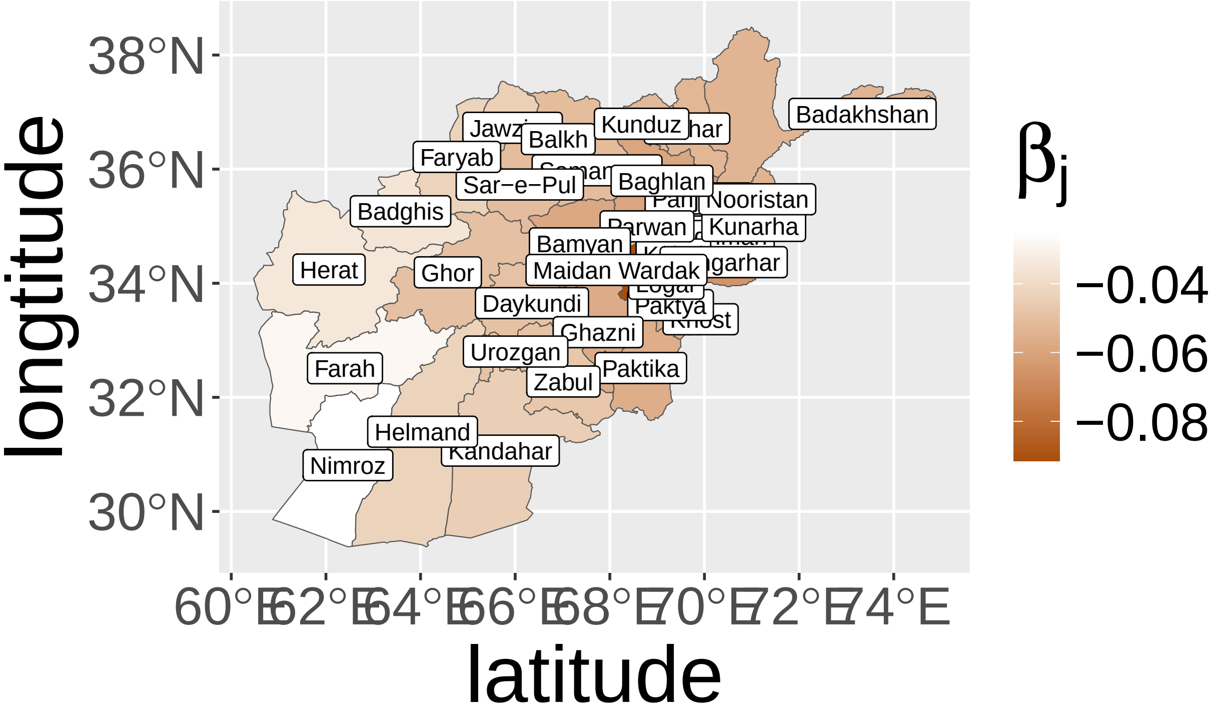


```
dev.print(pdf, paste(IMAGEOUT, "BETA.windex.rich.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_9 (Wealth - rich) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_windex_rich<- NULL
for (n in map$name){
    #if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rur
    #{
    #   significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    #}else{
    #   significant_gamma <- 0
    #}
    #beta_windex_rich<- append(beta_windex_rich, significant_gamma)
    beta_windex_rich<- append(beta_windex_rich, df$F[which(df$province==n & df$area=="Rural")])
}
map$beta_windex_rich<- beta_windex_rich
ggplot(map) +
    geom_sf(aes(fill = beta_windex_rich)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white"),
    values = scales::rescale(c(min(beta_windex_rich), max(beta_windex_rich[beta_windex_rich < 0]), 0)),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

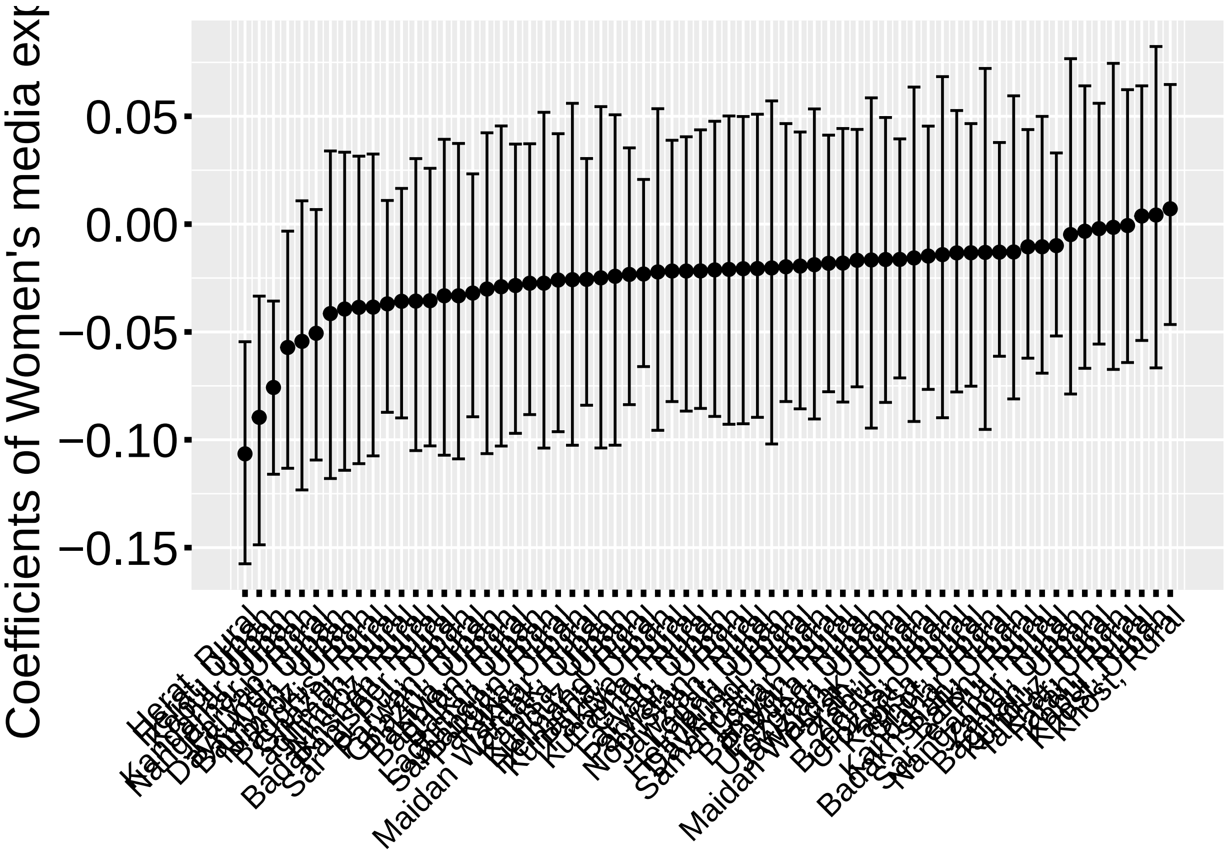


```r
dev.print(pdf, paste(IMAGEOUT, "BETA.windex.rich.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the posterior CI of gamma + theta coefficients of media

```r
i <- 1
GAMMA_media <- NULL
for (stratum in stratum_vector){
    gamma_ij <- NULL
    for (s in c(1:length(GAMMA.post))){
        gamma_ij <- c(gamma_ij, GAMMA.post[[s]][i, 10])
    }
    GAMMA_media <- rbind(GAMMA_media , gamma_ij)
    i <- i+1
}
qE<-apply(GAMMA_media+rep(THETA.post.mat[10,],66),1,quantile,probs=c(.025,.5,.975))
index <- order(qE[2,])

df <- data.frame(x = 1:66,
          province = stratum_province,
              area = stratum_area,
stratum_sample_size= stratum_sample_size,
                 F = qE[2,][index],
                 L = qE[1,][index],
                 U = qE[3,][index])
abc <- ggplot(df, aes(x = x, y = F)) +
  labs(y = "Coefficients of Women's media exposure", x = NULL) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = U, ymin = L)) +
  scale_x_continuous(breaks = c(1:66), labels = stratum_label[index]) +
  theme(axis.title=element_text(size=18), axis.text.x = element_text(size = 12, color = "black", angle =
        axis.ticks.x = element_line(color = "black", linewidth = 1),
        axis.text.y = element_text(size = 18, color = "black"),
        axis.ticks.y = element_line(color = "black", linewidth = 1))
print(abc)
```



```
dev.print(pdf, paste(IMAGEOUT, "BETA.media.pdf",sep=""),width=16,height=8)
```

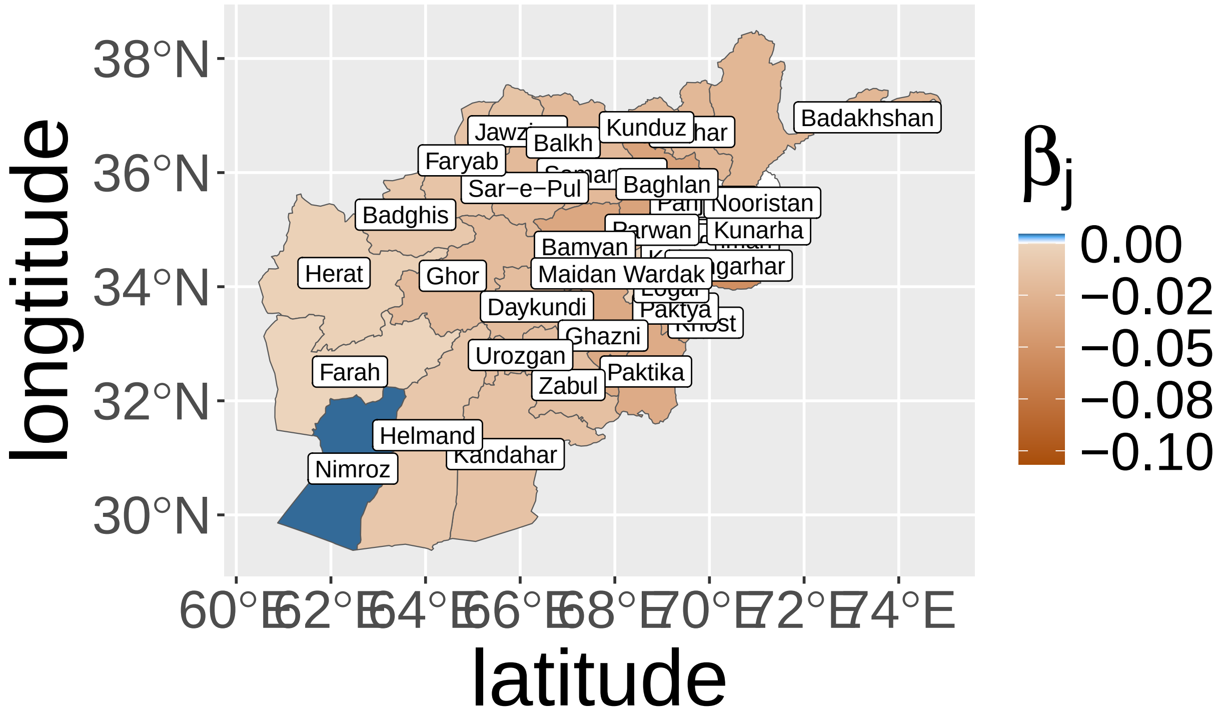
```
## pdf
##   2
```

```
write.csv(df, paste(IMAGEOUT,"qE.BETA.media.csv",sep=""))
```

```
# Coefficients of GAMMA_10 (media) + Theta at each province in urban area
map <- read_sf("data_src/map.json")
beta_media<- NULL
for (n in map$name){
    # In province Nooristan and Panjsher, there is no urban areas
    if (n == "Nooristan" | n == "Panjsher"){
        beta_media<- append(beta_media, 0)
    }else{
        #if (df$L[which(df$province==n & df$area=="Urban")] > 0 | df$U[which(df$province==n & df$area==
        #{
        #   significant_gamma <- df$F[which(df$province==n & df$area=="Urban")]
        #}else{
        #   significant_gamma <- 0
        #}
        #beta_media<- append(beta_media, significant_gamma)
        beta_media<- append(beta_media, df$F[which(df$province==n & df$area=="Urban")])
    }
}
# Add beta_media to the map data
map$beta_media<- beta_media

# Define the custom color scale
color_scale <- c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43")

# Define the values for the color scale
min_value <- min(beta_media, na.rm = TRUE)
max_negative_value <- max(beta_media[beta_media < 0], na.rm = TRUE)
min_positive_value <- min(beta_media[beta_media > 0], na.rm = TRUE)
max_value <- max(beta_media, na.rm = TRUE)

ggplot(map) +
    geom_sf(aes(fill = beta_media)) +
    scale_fill_gradientn(colors = color_scale,
    values = scales::rescale(c(min_value, max_negative_value, 0, min_positive_value, max_value)),
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm): collapsing to unique 'x' valu
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm): collapsing to unique 'x' valu
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm): collapsing to unique 'x' valu
```

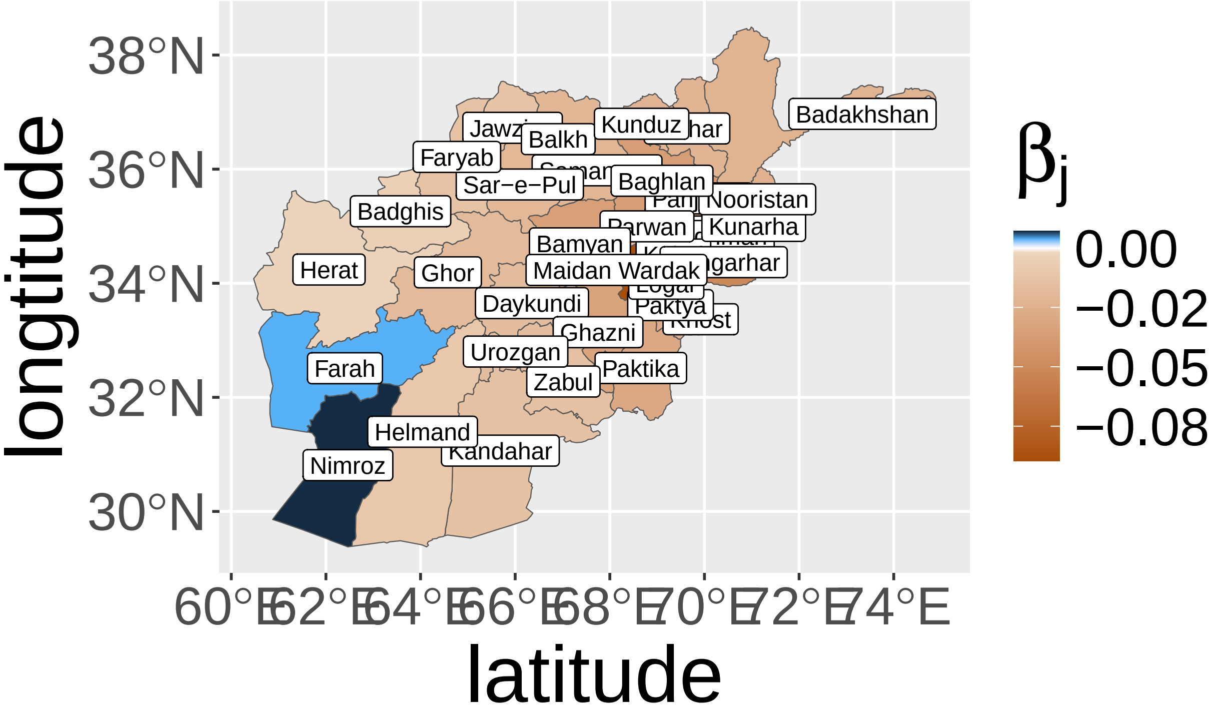


```
dev.print(pdf, paste(IMAGEOUT, "BETA.media.urban.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

```
# Coefficients of GAMMA_10 (media) + Theta at each province in rural area
map <- read_sf("data_src/map.json")
beta_media<- NULL
for (n in map$name){
    #if (df$L[which(df$province==n & df$area=="Rural")] > 0 | df$U[which(df$province==n & df$area=="Rur
    #{
    #   significant_gamma <- df$F[which(df$province==n & df$area=="Rural")]
    #}else{
    #   significant_gamma <- 0
    #}
    #beta_media<- append(beta_media, significant_gamma)
    beta_media<- append(beta_media, df$F[which(df$province==n & df$area=="Rural")])
}
map$beta_media<- beta_media
ggplot(map) +
    geom_sf(aes(fill = beta_media)) +
    scale_fill_gradientn(colors = c("#a94e08", "#ecd4bc", "white", "#56B1F7", "#132B43"),
    values = scales::rescale(c(min(beta_media), max(beta_media[beta_media< 0]),0, min(beta_media[beta_m
    guide = "colorbar", labels = function(x) sprintf("%.2f", x)) +
    geom_sf_label(aes(label = name), size = 3) +
    labs(fill=expression(paste(beta[j])), x="latitude", y="longtitude") +
    theme(
  legend.title = element_text(size = 30),     # Legend title size
  legend.text = element_text(size = 20),       # Legend item text size
    axis.title = element_text(size = 30),
    axis.text = element_text(size = 20)
)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not give correct results fo
## data
```



```r
dev.print(pdf, paste(IMAGEOUT, "BETA.media.rural.pdf",sep=""),width=16,height=16)
```

```
## pdf
##   2
```

To plot the POSTERIOR CI of Gamma of each stratum

```r
pdf(paste(IMAGEOUT,"GAMMA.post.pdf",sep=""), width=7, height=7)
par(mar=c(14,4.5,1,1),mgp=c(1.88,0.50,0))
for (strata in seq(1,m,1)){
    gammastrata <- NULL
    for (i in seq(1,(S-B)/thin,by=1)){
  # BETA.post[[i]] is the beta_j at ith simulation (thinned)
    gammastrata <- rbind(gammastrata, GAMMA.post[[i]][strata,])
    }
  # Credible Interval of beta_j at strata
    CIgammastrata <- apply(gammastrata, MARGIN=2, quantile, prob=c(0.025,0.5,0.975))
    Significant <- apply(CIgammastrata, MARGIN=2, function(q){return(ifelse(q[1]<0 & q[3]>0, FALSE, TRUE
    Meangammastrata<- apply(gammastrata, MARGIN=2, mean)
    CIgammastrata <- rbind(CIgammastrata, Significant, Meangammastrata)
    plot(1:q,type="n", ylab=expression(gamma[i]), xlab="", xlim=c(0,q+2), ylim=c(-4,3.5),xaxt="n", main=
  for (i in 1:q){
    color=ifelse(CIgammastrata[4,i],"red","gray")
    segments(i,CIgammastrata[1,i],i,CIgammastrata[3,i], lwd=2,
    col=color)
    #points(i, CItheta[1, i], pch=4, col=color)
    points(i, CIgammastrata[2, i], col=color)
    #points(i, CItheta[3, i], pch=4, col=color)
    }
    text(x = 1:q,
         y = par("usr")[3] - 0.1,
         labels = colnames(CIgammastrata),
         xpd = NA,
         srt = 45,
         cex = 1.2,
         adj = 1)
    abline(h=0, lty=2, col="GRAY")
}
dev.off()
```

```
## pdf
##   2
```

Dispersion Test

```
fit.mle <- glm(y~., data = data.frame(yX), family=poisson)
library(AER)
dispersiontest(fit.mle, alternative = "greater")
```

```
##
##  Overdispersion test
##
## data:  fit.mle
## z = -16.807, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.8585613
```

```
# Mean and variance for y in each stratum
nu_est <- rep(0,m)
for (strata in seq(1,m,1)){
cat("Stratum: ",strata)
y.perstratum <- y.list[[strata]]
cat(", length(y): ", length(y.perstratum))
X.perstratum <- X.list[[strata]]
yX.perstratum <- cbind(y.perstratum,X.perstratum)
cat(" ,mean: ")
cat(mean(y.perstratum))
cat(" ,var: ")
print(var(y.perstratum))
}
```

```
## Stratum:  1, length(y):  717 ,mean: 3.65272 ,var: [1] 6.648781
## Stratum:  2, length(y):  761 ,mean: 4.483574 ,var: [1] 9.757954
## Stratum:  3, length(y):  62 ,mean: 4.33871 ,var: [1] 6.522739
## Stratum:  4, length(y):  894 ,mean: 4.267338 ,var: [1] 8.800791
## Stratum:  5, length(y):  134 ,mean: 4.925373 ,var: [1] 9.077096
## Stratum:  6, length(y):  756 ,mean: 4.906085 ,var: [1] 10.32362
## Stratum:  7, length(y):  46 ,mean: 4.934783 ,var: [1] 7.395652
## Stratum:  8, length(y):  815 ,mean: 4.791411 ,var: [1] 10.81393
## Stratum:  9, length(y):  668 ,mean: 4.347305 ,var: [1] 9.546365
## Stratum:  10, length(y):  246 ,mean: 4.51626 ,var: [1] 6.879326
## Stratum:  11, length(y):  357 ,mean: 4.145658 ,var: [1] 6.967488
## Stratum:  12, length(y):  127 ,mean: 3.566929 ,var: [1] 5.88239
## Stratum:  13, length(y):  633 ,mean: 4.347551 ,var: [1] 9.312561
## Stratum:  14, length(y):  159 ,mean: 4.27673 ,var: [1] 7.365974
## Stratum:  15, length(y):  72 ,mean: 3.861111 ,var: [1] 7.191706
## Stratum:  16, length(y):  735 ,mean: 4.108844 ,var: [1] 8.843723
## Stratum:  17, length(y):  31 ,mean: 3.387097 ,var: [1] 5.511828
## Stratum:  18, length(y):  994 ,mean: 3.88833 ,var: [1] 7.722663
## Stratum:  19, length(y):  166 ,mean: 4.174699 ,var: [1] 8.41172
## Stratum:  20, length(y):  64 ,mean: 3.625 ,var: [1] 5.888889
## Stratum:  21, length(y):  907 ,mean: 4.113561 ,var: [1] 8.685765
## Stratum:  22, length(y):  149 ,mean: 4.818792 ,var: [1] 8.730455
## Stratum:  23, length(y):  655 ,mean: 3.864122 ,var: [1] 6.799552
## Stratum:  24, length(y):  1039 ,mean: 4.625602 ,var: [1] 8.671829
## Stratum:  25, length(y):  92 ,mean: 5.304348 ,var: [1] 10.9613
## Stratum:  26, length(y):  747 ,mean: 4.930388 ,var: [1] 11.23375
## Stratum:  27, length(y):  969 ,mean: 4.323013 ,var: [1] 9.65072
## Stratum:  28, length(y):  57 ,mean: 4.210526 ,var: [1] 9.56203
## Stratum:  29, length(y):  670 ,mean: 4.21194 ,var: [1] 8.511068
## Stratum:  30, length(y):  96 ,mean: 4.84375 ,var: [1] 11.35428
## Stratum:  31, length(y):  587 ,mean: 4.776831 ,var: [1] 10.17707
## Stratum:  32, length(y):  229 ,mean: 4.982533 ,var: [1] 9.973378
## Stratum:  33, length(y):  522 ,mean: 4.735632 ,var: [1] 10.84744
## Stratum:  34, length(y):  98 ,mean: 4.173469 ,var: [1] 7.814959
## Stratum:  35, length(y):  668 ,mean: 3.523952 ,var: [1] 5.853998
## Stratum:  36, length(y):  260 ,mean: 3.915385 ,var: [1] 6.703237
## Stratum:  37, length(y):  408 ,mean: 4.151961 ,var: [1] 9.200438
## Stratum:  38, length(y):  57 ,mean: 3.807018 ,var: [1] 7.837093
## Stratum:  39, length(y):  831 ,mean: 3.799037 ,var: [1] 7.946312
## Stratum:  40, length(y):  140 ,mean: 3.657143 ,var: [1] 5.334841
## Stratum:  41, length(y):  52 ,mean: 4.326923 ,var: [1] 6.381222
## Stratum:  42, length(y):  722 ,mean: 3.736842 ,var: [1] 5.739251
## Stratum:  43, length(y):  52 ,mean: 2.826923 ,var: [1] 4.498869
## Stratum:  44, length(y):  690 ,mean: 3.92029 ,var: [1] 8.57564
## Stratum:  45, length(y):  722 ,mean: 3.896122 ,var: [1] 7.277683
## Stratum:  46, length(y):  93 ,mean: 4.225806 ,var: [1] 9.568022
## Stratum:  47, length(y):  669 ,mean: 4.055306 ,var: [1] 9.815799
## Stratum:  48, length(y):  112 ,mean: 5.544643 ,var: [1] 11.54754
## Stratum:  49, length(y):  783 ,mean: 4.545338 ,var: [1] 12.15363
## Stratum:  50, length(y):  396 ,mean: 4.752525 ,var: [1] 10.53607
## Stratum:  51, length(y):  688 ,mean: 5.167151 ,var: [1] 14.51496
## Stratum:  52, length(y):  243 ,mean: 4.069959 ,var: [1] 7.123185
## Stratum:  53, length(y):  681 ,mean: 4.088106 ,var: [1] 8.483402
## Stratum:  54, length(y):  120 ,mean: 4.016667 ,var: [1] 7.310644
## Stratum:  55, length(y):  692 ,mean: 4.257225 ,var: [1] 9.655881
## Stratum:  56, length(y):  127 ,mean: 5.188976 ,var: [1] 11.36083
## Stratum:  57, length(y):  795 ,mean: 5.046541 ,var: [1] 15.1603
## Stratum:  58, length(y):  50 ,mean: 4.04 ,var: [1] 9.100408
## Stratum:  59, length(y):  704 ,mean: 4.019886 ,var: [1] 6.710842
## Stratum:  60, length(y):  340 ,mean: 3.726471 ,var: [1] 9.473634
## Stratum:  61, length(y):  692 ,mean: 3.949422 ,var: [1] 10.49382
## Stratum:  62, length(y):  52 ,mean: 4.980769 ,var: [1] 8.176094
## Stratum:  63, length(y):  52 ,mean: 3.942308 ,var: [1] 8.643665
## Stratum:  64, length(y):  651 ,mean: 4.360983 ,var: [1] 9.351029
## Stratum:  65, length(y):  190 ,mean: 4.268421 ,var: [1] 7.784712
## Stratum:  66, length(y):  577 ,mean: 4.55286 ,var: [1] 10.07402
```

```r
# Dispersion test for each stratum
nu_est <- rep(0,m)
for (strata in seq(1,m,1)){
cat("Stratum: ",strata)
y.perstratum <- y.list[[strata]]
cat(", length(y): ", length(y.perstratum))
X.perstratum <- X.list[[strata]]
yX.perstratum <- cbind(y.perstratum,X.perstratum)
fit.mle.perstratum <- glm(y.perstratum~., data = data.frame(yX.perstratum), family=poisson)
test.perstratum <- dispersiontest(fit.mle.perstratum, alternative = "less")
cat(", dispersion: ")
cat(test.perstratum$estimate, ", p-value: ")
print(test.perstratum$p.value)
nu_est[strata] <- 1/test.perstratum$estimate
#fit.glm.cmp.perstratum <- glm.cmp(formula = y.perstratum~., data = data.frame(yX.perstratum))
}
```

```
## Stratum:  1, length(y):  717, dispersion: 0.7478419 , p-value: [1] 1.412808e-09
## Stratum:  2, length(y):  761, dispersion: 0.7706786 , p-value: [1] 5.059217e-07
## Stratum:  3, length(y):  62, dispersion: 0.4543743 , p-value: [1] 4.635385e-07
## Stratum:  4, length(y):  894, dispersion: 0.8444825 , p-value: [1] 0.0001637626
## Stratum:  5, length(y):  134, dispersion: 0.7138425 , p-value: [1] 0.0006396543
## Stratum:  6, length(y):  756, dispersion: 0.6888515 , p-value: [1] 3.765817e-14
## Stratum:  7, length(y):  46, dispersion: 0.5790619 , p-value: [1] 2.524345e-05
## Stratum:  8, length(y):  815, dispersion: 0.8660918 , p-value: [1] 0.002884009
## Stratum:  9, length(y):  668, dispersion: 0.9410398 , p-value: [1] 0.1727588
## Stratum:  10, length(y):  246, dispersion: 0.6435486 , p-value: [1] 1.0321e-07
## Stratum:  11, length(y):  357, dispersion: 0.8443313 , p-value: [1] 0.009510595
## Stratum:  12, length(y):  127, dispersion: 0.6094665 , p-value: [1] 1.525838e-06
## Stratum:  13, length(y):  633, dispersion: 0.7570317 , p-value: [1] 4.002447e-08
## Stratum:  14, length(y):  159, dispersion: 0.6371924 , p-value: [1] 2.614817e-05
## Stratum:  15, length(y):  72, dispersion: 0.553594 , p-value: [1] 0.0002074724
## Stratum:  16, length(y):  735, dispersion: 0.8563992 , p-value: [1] 0.004844941
## Stratum:  17, length(y):  31, dispersion: 0.3755772 , p-value: [1] 6.820835e-10
## Stratum:  18, length(y):  994, dispersion: 0.8579531 , p-value: [1] 0.0002320519
## Stratum:  19, length(y):  166, dispersion: 0.8354493 , p-value: [1] 0.02416315
## Stratum:  20, length(y):  64, dispersion: 0.6418313 , p-value: [1] 0.002400555
## Stratum:  21, length(y):  907, dispersion: 0.9213733 , p-value: [1] 0.04694516
## Stratum:  22, length(y):  149, dispersion: 0.9279566 , p-value: [1] 0.2810104
## Stratum:  23, length(y):  655, dispersion: 0.6767119 , p-value: [1] 1.297952e-13
## Stratum:  24, length(y):  1039, dispersion: 0.6884197 , p-value: [1] 2.35733e-20
## Stratum:  25, length(y):  92, dispersion: 0.7543614 , p-value: [1] 0.01606798
## Stratum:  26, length(y):  747, dispersion: 0.8727541 , p-value: [1] 0.005988717
## Stratum:  27, length(y):  969, dispersion: 0.9039041 , p-value: [1] 0.02040082
## Stratum:  28, length(y):  57, dispersion: 0.6457024 , p-value: [1] 0.006671986
## Stratum:  29, length(y):  670, dispersion: 0.89964 , p-value: [1] 0.03073061
## Stratum:  30, length(y):  96, dispersion: 0.5373864 , p-value: [1] 1.634232e-06
## Stratum:  31, length(y):  587, dispersion: 0.7968513 , p-value: [1] 8.948722e-05
## Stratum:  32, length(y):  229, dispersion: 0.7069118 , p-value: [1] 6.725559e-05
## Stratum:  33, length(y):  522, dispersion: 0.936771 , p-value: [1] 0.1658696
## Stratum:  34, length(y):  98, dispersion: 0.8208013 , p-value: [1] 0.07593779
## Stratum:  35, length(y):  668, dispersion: 0.8038829 , p-value: [1] 3.073429e-05
## Stratum:  36, length(y):  260, dispersion: 0.6132919 , p-value: [1] 1.613648e-09
## Stratum:  37, length(y):  408, dispersion: 0.8298544 , p-value: [1] 0.007050458
## Stratum:  38, length(y):  57, dispersion: 0.7116819 , p-value: [1] 0.007915491
## Stratum:  39, length(y):  831, dispersion: 0.8484104 , p-value: [1] 0.0005343439
## Stratum:  40, length(y):  140, dispersion: 0.6889147 , p-value: [1] 0.000335265
## Stratum:  41, length(y):  52, dispersion: 0.6905925 , p-value: [1] 0.0284612
## Stratum:  42, length(y):  722, dispersion: 0.7594377 , p-value: [1] 1.223856e-08
## Stratum:  43, length(y):  52, dispersion: 0.5935051 , p-value: [1] 0.0006639758
## Stratum:  44, length(y):  690, dispersion: 0.7266533 , p-value: [1] 5.553565e-09
## Stratum:  45, length(y):  722, dispersion: 0.8189436 , p-value: [1] 0.0003890857
## Stratum:  46, length(y):  93, dispersion: 0.8109532 , p-value: [1] 0.04060041
## Stratum:  47, length(y):  669, dispersion: 1.049244 , p-value: [1] 0.7961684
## Stratum:  48, length(y):  112, dispersion: 0.854164 , p-value: [1] 0.08689653
## Stratum:  49, length(y):  783, dispersion: 0.9392 , p-value: [1] 0.1425503
## Stratum:  50, length(y):  396, dispersion: 0.8038884 , p-value: [1] 0.001881773
## Stratum:  51, length(y):  688, dispersion: 1.090696 , p-value: [1] 0.888008
## Stratum:  52, length(y):  243, dispersion: 0.7598957 , p-value: [1] 0.0001593424
## Stratum:  53, length(y):  681, dispersion: 0.7342408 , p-value: [1] 2.781894e-08
## Stratum:  54, length(y):  120, dispersion: 0.7566334 , p-value: [1] 0.0238376
## Stratum:  55, length(y):  692, dispersion: 0.9232582 , p-value: [1] 0.08888097
## Stratum:  56, length(y):  127, dispersion: 0.6868704 , p-value: [1] 0.001396002
## Stratum:  57, length(y):  795, dispersion: 0.9221721 , p-value: [1] 0.06817484
## Stratum:  58, length(y):  50, dispersion: 0.8281621 , p-value: [1] 0.1868348
## Stratum:  59, length(y):  704, dispersion: 0.7414088 , p-value: [1] 5.205096e-08
## Stratum:  60, length(y):  340, dispersion: 0.8320685 , p-value: [1] 0.004555804
## Stratum:  61, length(y):  692, dispersion: 0.8431907 , p-value: [1] 0.001754155
## Stratum:  62, length(y):  52, dispersion: 0.6382703 , p-value: [1] 0.006401623
## Stratum:  63, length(y):  52, dispersion: 0.5695501 , p-value: [1] 0.0002286428
## Stratum:  64, length(y):  651, dispersion: 0.7324714 , p-value: [1] 2.805566e-08
## Stratum:  65, length(y):  190, dispersion: 0.6047478 , p-value: [1] 4.450887e-08
## Stratum:  66, length(y):  577, dispersion: 0.766722 , p-value: [1] 4.657078e-06
```

```
pdf(paste(IMAGEOUT,"CEB_Freq.pdf",sep=""), width=10)
par(mar=c(4,4.5,1,1),mgp=c(1.88,0.50,0))
barplot(table(y), ylab="Frequency",
        xlab="Number of children ever born", cex.lab = 2, names.arg=c(0:18), font.axis = 2.5)
dev.off()
```

```
## pdf
##   2
```

```
# BIC_h for the Hierarchical Model
n<-length(y)
MLE <- 0
for(j in 1:m)
{
MLE <- MLE + sum(wgt.list[[j]] * dpois(y.list[[j]], lambda=exp(matmul(Z.list[[j]], apply(GAMMA.post.mat
}
BIC_h <- q * log(m) + (p-q) * log(n) - 2 * MLE
print(BIC_h)
```

```
## [1] 110743.9
```

```
if(!require(mvtnorm)) install.packages("mvtnorm", dependencies=TRUE)
library(mvtnorm)

# Density function for spike and slab prior
# CSS for Continuous Spike-and-Slab

# dCSS is the density function for the spike-and-slab prior, where x is the input value, df_spike and d
dCSS <- function(x, df_spike, location_spike, scale_spike, df_slab, location_slab, scale_slab, p_spike)
  spike_density <- dt((x - location_spike) / scale_spike, df_spike) / scale_spike
  slab_density <- dt((x - location_slab) / scale_slab, df_slab) / scale_slab
  p_spike * spike_density + (1 - p_spike) * slab_density
}


# Cumulative distribution function for spike and slab prior
pCSS <- function(q, df_spike, location_spike, scale_spike, df_slab, location_slab, scale_slab, p_spike)
  spike_cdf <- pt((q - location_spike) / scale_spike, df_spike)
  slab_cdf <- pt((q - location_slab) / scale_slab, df_slab)
  p_spike * spike_cdf + (1 - p_spike) * slab_cdf
}

# Quantile function for spike and slab prior
qCSS <- function(p, df_spike, location_spike, scale_spike, df_slab, location_slab, scale_slab, p_spike)
  quantile_function <- function(q) {
    pCSS (q, df_spike, location_spike, scale_spike, df_slab, location_slab, scale_slab, p_spike) - p
  }
  uniroot(quantile_function, c(-1e6, 1e6))$root
}

# Example parameters
df_spike <- 2
location_spike <- 0
scale_spike <- 0.2
df_slab <- 2
location_slab <- 0
scale_slab <- 2
p_spike <- 0.8

s <- seq(0.00001, 6, by = 0.00001)
for(i in 1:length(s)){
  q <- qCSS(0.975, df_spike , location_spike, s[i], df_slab , location_slab, 1/s[i], p_spike)
  if(q < 6){
    cat("q = ", q, "\n")
    cat("sd = ", s[i], "\n")
    break
  }
}
pCSS(6, df_spike , location_spike, s[i], df_slab , location_slab, 1/s[i], p_spike)

# Generate values for plotting
x <- seq(-6, 6, by = 0.0001)
ys <- dCSS(x, df_spike, location_spike, 0.2742, df_slab, location_slab, 1/0.2742, p_spike)
```

```
dDSS <- function(theta, pi, mu, sigma) {
    # Spike component (Dirac delta function at theta = 0)
    spike <- pi * (theta == 0)

    # Slab component (Normal distribution)
```

```r
    slab <- (1 - pi) * dnorm(theta, mean = mu, sd = sigma)

    # Combine spike and slab
    density <- spike + slab
    return(density)
}

pDSS <- function(q, pi, mu, sigma) {
    if (q < mu) {
        return((1 - pi) * pnorm(q, mean = mu, sd = sigma))
  } else {
        return(pi + (1 - pi) * pnorm(q, mean = mu, sd = sigma))
    }
}

qDSS <- function(p, pi, mu, sigma) {
  if (p <= (1 - pi) * 0.5) {
    return(mu + sigma * qnorm( p / (1 - pi)))
  } else if ( p >= pi + (1 - pi) * 0.5) {
    return(mu + sigma * qnorm((p - pi) / (1 - pi)))
  } else{
        return (mu)
    }
}
pmn.theta <- rep(0,p)
psd.theta<- rep(0,p)
pi=0.3

s <- seq(0.00001, 6, by = 0.00001)
for(i in 1:length(s)){
  q <- qDSS(0.975, pi, 0, s[i] )
  if(q > 6){
    cat("q = ", q, "\n")
    cat("sd = ", s[i], "\n")
    break
  }
}
```

```
## q =  6.000016
## sd =  3.32827
```

```r
pDSS(q=-6, pi, mu=0, sigma=3.32827)
```

```
## [1] 0.02500026
```

```r
pDSS(q=6, pi, mu=0, sigma=3.32827)
```

```
## [1] 0.9749997
```

```r
x <- seq(-6, 6, by = 0.0001)
yd <- dDSS(theta=x, pi=pi, mu=0, sigma=3.32827)
```

```r
library(extraDistr)

# Intercept
s <- seq(0.00001, 6, by = 0.00001)
```

```r
for(i in 1:length(s)){
  q <- qnorm(0.975, mean = 0, sd = s[i])
  if(q > 6){
    cat("q = ", q, "\n")
    cat("sd = ", s[i], "\n")
    break
  }
}
pnorm(-6, mean = 0, sd = 3.06129)
pnorm(6, mean = 0, sd = 3.06129)


for(i in 1:length(s)){
  q <- qlaplace(0.975, mu = 0, sigma = s[i])
  if(q > 6){
    cat("q = ", q, "\n")
    cat("sd = ", s[i], "\n")
    break
  }
}
plaplace(-6, mu = 0, sigma = 2.00285)
plaplace(6, mu = 0, sigma = 2.00285)


for(i in 1:length(s)){
  q <- qcauchy(0.975, location = 0, scale = s[i])
  if(q > 6){
    cat("q = ", q, "\n")
    cat("sd = ", s[i], "\n")
    break
  }
}
pcauchy(-6, location = 0, scale = 0.47222)
pcauchy(6, location = 0, scale = 0.47222)


x <- seq(-6, 6, by = 0.0001)

yn <- dnorm(x, mean = 0, sd = 3.06129)
yl <- dlaplace(x, mu = 0, sigma = 2.00285)
yc <- dcauchy(x, location = 0, scale = 0.47222)

par(mar=c(4,4.5,1,1),mgp=c(1.88,0.50,0))
plot(x, ys, type = "l", lty = 1, ylab = "Prior densities", xlab = "", cex.lab = 2, col = "red",lwd=3)
lines(x, yl, lty = 1,lwd = 3, col = "orange")
lines(x, yn, lty = 1,lwd = 3, col = "blue")
#lines(x, yc, lty = 1,lwd = 3, col = "green")
#lines(x, yd, lty = 1,lwd = 3, col = "cyan")
#abline(v = -2.835)
legend(-3, 1, c(
    "Spike and slab (C) (0, 0.2742)",
    "Normal (0, 3.06129)",
    "Laplace (0, 2.00285)",
    "Cauchy (0, 0.47222)",
    #"Spike and slab (D) (0,pi=0.3,3.32827)"
    ), lty = c(1, 1, 1, 1),lwd = c(3, 3, 3, 3), col = c("red", "blue", "orange", "green", "cyan"), bty
dev.print(pdf, paste(IMAGEOUT, "Prior_Intercept.pdf",sep=""),width=8,height=8)
```

```
library(mpcmp)
fit.glm.cmp <- glm.cmp(formula = y ~ ., data = data.frame(yX))
```

```
library(VGAM)
gdata <- data.frame(x2 = runif(nn <- 500))
gdata <- transform(gdata, y1 = rgenpois2(nn, exp(2 + x2),
                               loglink(-1, inverse = TRUE)))
gfit2 <- vglm(y1 ~ x2, genpoisson2, gdata, trace = TRUE)
gfitgp2 <- vglm(y ~ ., genpoisson2, data.frame(yX), trace=FALSE)
coef(gfitgp2, matrix = TRUE)
summary(gfit2)
```

```
theta0.hat <- NULL
for (strata in seq(1,m,1)){
cat("Stratum: ",strata)
y.perstratum <- y.list[[strata]]
cat(", length(y): ", length(y.perstratum))
X.perstratum <- X.list[[strata]]
yX.perstratum <- cbind(y.perstratum,X.perstratum)
fit.mle.perstratum <- glm(y.perstratum~women_age+women_age2+HH_age+HH_age2+women_agem+women_edu+HH_edu+
theta0.hat <- rbind(theta0.hat,summary(fit.mle.perstratum)$coefficients[,1])
}
theta0.hat.cov<-cov(theta0.hat)
```

```
library("glmmTMB")
df.yX <- as.data.frame(cbind(yX,df_sel$stratum))
```

```
fit_poisson <- glmmTMB(formula = y~women_age+women_age2+HH_age+women_agem+women_edu+HH_edu+windex3.Poor
RESULTPATH = "./data_output/HP/"
save(fit_poisson, file=paste(RESULTPATH,"fit_poisson.RData",sep=""))
```

```
RESULTPATH = "./data_output/HP/"
load(paste(RESULTPATH,"fit_poisson.RData",sep=""))
sink(paste(RESULTPATH,"HP.summary.txt"))
summary(fit_poisson)
sink()
as.matrix(exp(summary(fit_poisson)$coeff$cond[,1]))
```