

Regional-Based Mitigation to Reduce Wildlife-Vehicle Collisions

- Supplementary code

Snow, N. P., Zhang, Z., Finley, A. O., Rudolph, B. A., Porter, W. F., Williams, D. M. and Winterstein, S. R.

December 12, 2017

Contents

Data Processing	1
Visualize spatial residuals from model comparison	4
Visualize the estimated effects	6
Plot the biological effects for the influential variables on raw scale	8

Data Processing

Load necessary packages:

```
rm(list=ls())
require(maptools)
require(maps)
require(fields)
require(lattice)
require(ggplot2)
require(R.matlab)
require(rgdal)
require(RColorBrewer)
require(Hmisc) #xYplot Cbind
windowsFonts(Times=windowsFont("Times New Roman"))
trellis.par.set(grid.pars = list(fontfamily = "Times"))
saveFig <- FALSE #save figure to files

dir0 <- c('./data/') #directory for shape files and adjacency matrix
dat <- read.csv(paste0('./','Deer_Data_Master2.csv'))
table(dat$STATE, dat$ECOZONE) # nonnested

##      agriculture agriculture_forest northern_forest
##      IA          80            19            0
##      IL          57            45            0
##      MI           0            36            47
##      WI           0            40            31

length(unique(paste(dat$STATE, dat$COUNTY))) # 355

## [1] 355
shape <- readShapePoly(paste0(dir0,"shapefile/Counties_for_analysis_region.shp"))

## Warning: use rgdal::readOGR or sf::st_read
county.dat <- paste(dat$STATE, dat$COUNTY, sep=' - ')
county.shape <- paste(shape$STATE, shape$COUNTY, sep=' - ')
all(county.dat == county.shape) # dat match with shape

## [1] TRUE
```

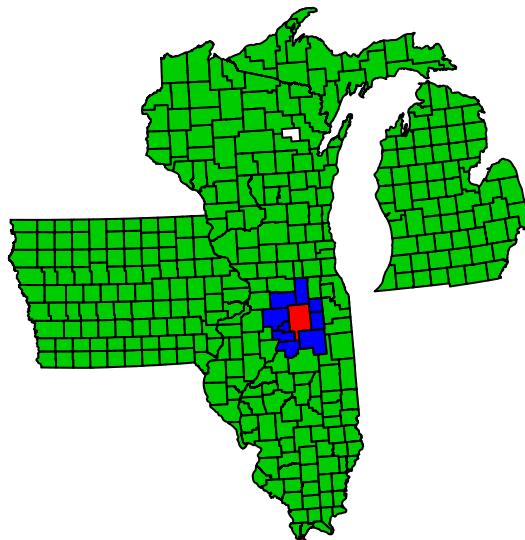
```

# need to save Adj csv from txt; Now match Adj.csv with Num.txt
W0 <- read.csv(paste0(dir0,'Adjacency/Adj.csv'), head=F)
M0 <- read.csv(paste0(dir0,'Adjacency/Num.txt'), head=F)
all(apply(W0, 1, function(x) return(sum(!is.na(x)))) == M0[,1])

## [1] TRUE
n <- nrow(W0)
W <- matrix(0,n,n); for(i in 1:n) W[i,na.omit(as.numeric(W0[i,]))] <- 1
# check the adjacency matrix
all(W==t(W))

## [1] TRUE
cols <- rep(3,n); k <- 149; cols[c(k,na.omit(as.numeric(W0[k,])))] <- 4; cols[k] <- 2
par(mar=c(0,0,0,0)); plot(shape,col=cols) # shape match with W

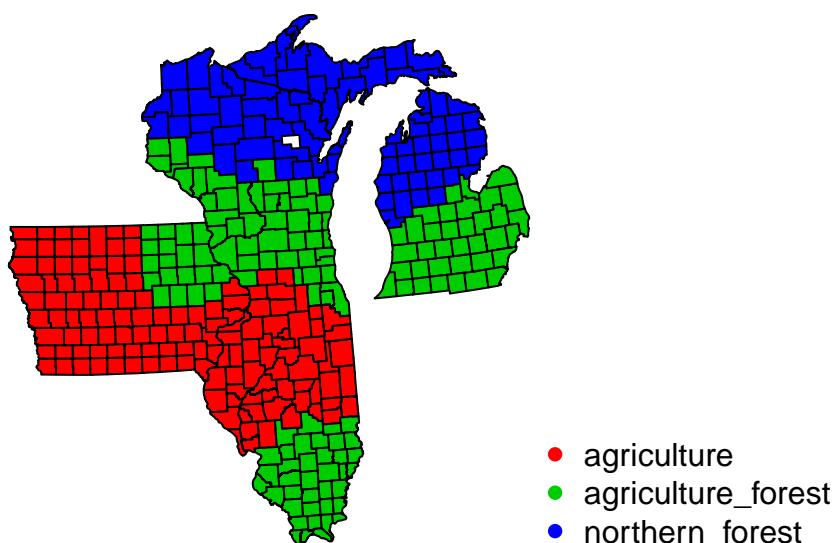
```



```

# visualize ecozone
par(mar=c(0,0,0,0)); plot(shape,col=as.integer(dat$ECOZONE)+1) # shape match with W
legend('bottomright', legend=levels(dat$ECOZONE), col=c(1:3)+1, pch=16, bty='n')

```



```

Y <- dat[,2+c(5:16)]
J <- ncol(Y) # 12 years
inds <- seq(7,ncol(dat),by=J)

```

```

nvar <- length(inds)
varname <- c('DVC','AVMT','Antler','CONT','P_AG')
varlabs <- c('Traffic','Abundance of deer','Contagion','Proportion of agriculture','Traffic squared',
           'Traffic x Abundance','Contagion x Abundance','STATE: IL','STATE: MI','STATE: WI',
           'ECOZONE: agriculture_forest','ECOZON: Enorthern_forest')

par(mfrow=c(2,3),mar=c(2,2,2,0))
ys <- matrix(NA, n*J, nvar)
ss <- as.list(rep(NA, nvar)) #summary statistics for raw data
names(ss) <- c('DVC', varlabs[1:4]) #varname
nvar <- length(inds <- seq(7,ncol(dat),by=J) )
for(j in 1:nvar){
  y <- dat[,inds[j]:(inds[j]+J-1)]
  ss[[j]] <- rbind(apply(y, 2, mean), apply(y, 2, sd), apply(y, 2, range))
  if(j==2) ss[[j]][4,] <- apply(y, 2, quantile,.9) #max -> quantile since highly skewed
  row.names(ss[[j]]) <- c('mean','stdev','min','max')
  if(j>1) y <- scale(y) #year-wise standardized covarites (j=1: response)
  #scale the 4 raw predictors
  palette(topo.colors(J)) #brewer.pal(J,'Set3')
  matplot(t(y), type='n', main=varname[j])
  for(i in 1:nrow(y)) lines(1:J,y[i,],lty=1,col=i)
  ys[,j] <- as.numeric(as.matrix(y))
}
indT <- as.numeric(rep(1:J, each=n)) #depends on the organization of y (data)

# extract design matrix
dat.lm <- as.data.frame(ys); names(dat.lm) <- varname
summary(dat.lm)

```

```

##      DVC          AVMT          Antler          CONT
##  Min.   : 0.0  Min.   :-0.35751  Min.   :-1.1145  Min.   :-2.25685
##  1st Qu.: 101.0 1st Qu.:-0.28591  1st Qu.:-0.7188  1st Qu.:-0.83938
##  Median : 196.5 Median :-0.23005  Median :-0.4150  Median :-0.05214
##  Mean   : 325.8 Mean   : 0.00000  Mean   : 0.0000  Mean   : 0.00000
##  3rd Qu.: 408.0 3rd Qu.:-0.06653  3rd Qu.: 0.5567  3rd Qu.: 0.74510
##  Max.   :2450.0  Max.   :14.30315  Max.   : 4.8685  Max.   : 2.20014
##      P_AG
##  Min.   :-2.0479
##  1st Qu.:-0.6265
##  Median : 0.2512
##  Mean   : 0.0000
##  3rd Qu.: 0.7930
##  Max.   : 1.3339

dat.lm$ECOZONE <- rep(dat$ECOZONE, J)
dat.lm$STATE <- rep(dat$STATE, J)
ys2 <- cbind(dat.lm$DVC, model.matrix(DVC ~ AVMT+Antler+CONT+P_AG+I(AVMT^2)+AVMT*Antler+CONT*Antler,
                                         data=dat.lm)[,-1])
ys2 <- cbind(ys2, model.matrix(DVC ~ STATE+ECOZONE, data=dat.lm)[,-1])
# cor(ys2[,-1]) #check collinearity
# change E (offset) to be the Area
E <- rep(dat$Area_KMSQ, J) #E <- rep(dat$RD_DENSITY, J) # plot(E, ys[,1])
ezone <- as.integer(dat.lm$ECOZONE) # table(ezone, dat.lm$ECOZONE)
ezone0 <- as.integer(dat$ECOZONE) # table(ezone0, dat$ECOZONE)
uezone <- levels(dat$ECOZONE)

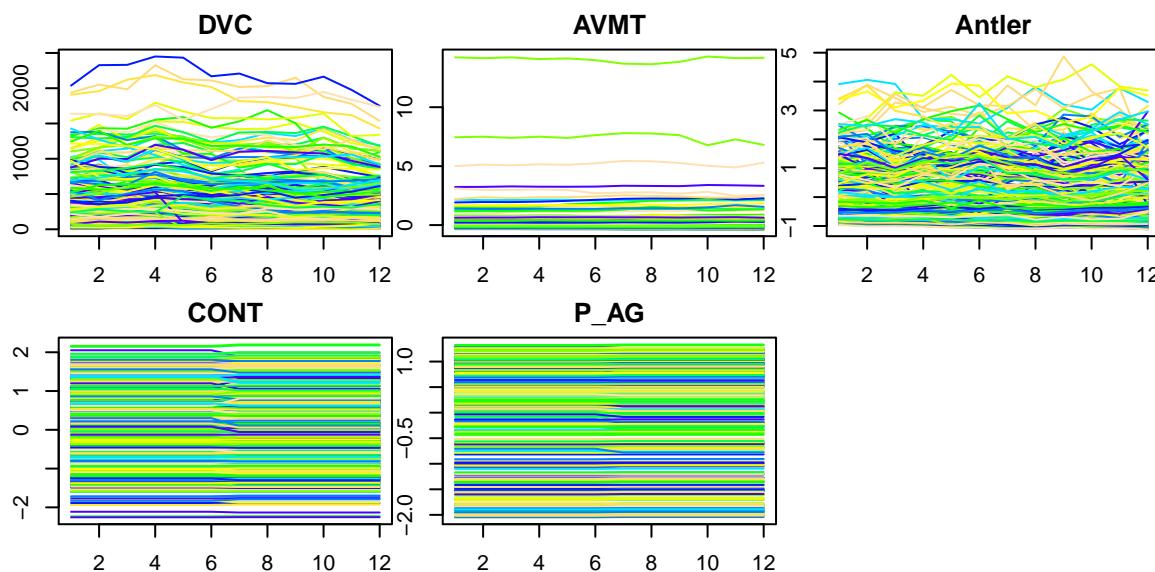
# get training set and validation set
tab <- table(dat$STATE)
num_state <- as.integer(dat$STATE) # table(dat$STATE, num_state)
nstate <- length(ustate <- unique(dat$STATE))

```

```

set.seed(25)
tvind <- list(nbin <- 10)
for(j in 1:nstate){
  if(j == 1) {for(i in 1:nbin) tvind[[i]] <- integer() }
  m <- length(a <- which(dat$STATE == ustate[j]))
  a <- a[sample(1:m)] #random permutation
  ind <- round(seq(1,m,len=nbin+1));
  ind[1] <- 1; ind[length(ind)] <- m+1
  tmp <- integer()
  for(i in 1:nbin){
    tmp <- c(tmp, a[ind[i]:(ind[i+1]-1)])
    tvind[[i]] <- c(t vind[[i]], a[ind[i]:(ind[i+1]-1)])
  }
}
nrows <- max(unlist(lapply(t vind, 'length' )))
t vmat <- matrix(0, nrows, nbin)
for(i in 1:nbin) t vmat[1:length(t vind[[i]]),i] <- tvind[[i]]

```



```

##### save data for MATLAB for modeling part
writeMat('./t vmat.mat', t vmat = t vmat)
writeMat('./datAll.mat', ys=ys2, W=W, E=E, ezone=as.numeric(ezone),
         ezone0=as.numeric(ezone0), indT=indT, uezone=uezone)

```

We next proceed with Markov chain Monte Carlo (MCMC) technique for the produced data. See `{spDynamPois.m}` for detailed model implementation.

Visualize spatial residuals from model comparison

```

dat1 <- readOGR(dsn=paste(dir0,"shapefile",sep=''),"Counties_for_analysis_region")
## OGR data source with driver: ESRI Shapefile
## Source: "./data/shapefile", layer: "Counties_for_analysis_region"
## with 355 features
## It has 66 fields
dat1 <- spTransform(dat1, CRS("+init=epsg:4267")) #27700
ID <- formatC(dat1$ID, width = 2, flag = "0")
dat1 <- spChFIDs(dat1, ID)
bluepal <- colorRampPalette(c("azure1", "steelblue4"))

```

```

tmp <- readMat('./paras.mat')
Yhat <- tmp$Yhatmat[,2]  # # model 2 is the best
Yobs <- tmp$Yobs
a <- cbind(Yhat, Yobs)
mean(Yobs>=Yhat[,2] & Yobs<=Yhat[,3]) #coverage% for observed Y (DVC)

## [1] 0.9267606

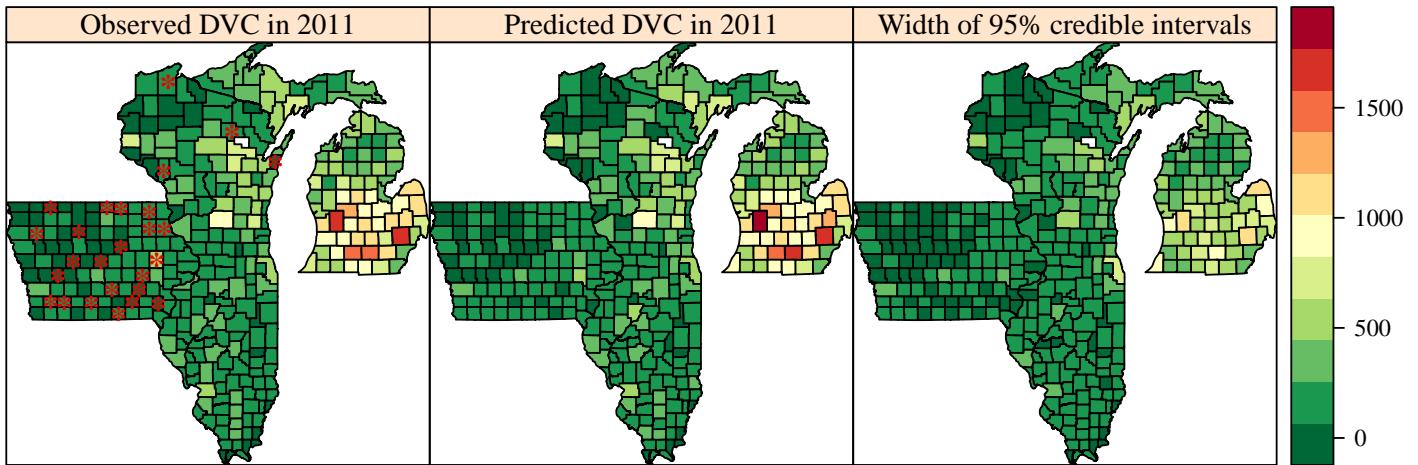
inds <- which(Yobs<Yhat[,2] | Yobs>Yhat[,3]) #a[inds, ]

dat1$yobs <- Yobs #tmp$LRR.obs
dat1$yest <- Yhat[,1] #tmp$LRR.hat[,2] # model 2 is the best
dat1$yestL <- Yhat[,2] #tmp$LRR.hat[,2] # model 2 is the best
dat1$yestU <- Yhat[,3] #tmp$LRR.hat[,2] # model 2 is the best
dat1$LU <- dat1$yestU - dat1$yestL
dat1$se <- Yhat[,4]

cols0 <- c('#a50026', '#d73027', '#f46d43', '#fdae61', '#fee08b', '#ffffbf', '#d9ef8b',
           '#a6d96a', '#66bd63', '#1a9850', '#006837') #brewer.pal(11, 'Spectral')
cols0 <- cols0[length(cols0):1]
coords <- coordinates(dat1)
xy <- coords[inds,]; xy[,2] <- xy[,2]-.13; xy[,1] <- xy[,1]+.01
locs <- list("sp.text", xy, rep("*",length(inds)), font=2, cex=1.2, which=1, col='red2')
#locs <- list("sp.points", coords[inds,], pch=16, col='red', cex=1, which=1)

if(saveFig) pdf('spatialMap_cv10_dvc_v4.pdf', width = 8.2, height = 4, pointsize=16)
lattice.options(default.theme = canonical.theme(color = FALSE))
spplot(dat1, c("yobs", "yest", "LU"), cuts=10, col.regions = cols0, layout=c(3,1), as.table=T, sp.layout=locs,
       names.attr=c("Observed DVC in 2011", "Predicted DVC in 2011", "Width of 95% credible intervals")
) #bluepal(16))

```



```
if(saveFig) dev.off()
```

Visualize the estimated effects

```

varlabs1 <- c('Intercept', varlabs)
p <- 5; P0 <- 8
nt <- length( years <- c(2000:2011) )
ns <- 3 #number of states (IL, MI, WI) excluding the reference (IA)
nzone <- length(uzone <- c('Agriculture','Forest-Agriculture Matrix','Northern Forest'))
# note state effect is not varying across ecozones
varlabs0 <- c(rep(c(rep(varlabs1[1:p],nzone), varlabs1[P0+c(1:ns)]), nt),
             rep(c('variation', 'spatial dependence'), each=nt))
uezonenew <- c(rep(c(rep(uzone,each=p),
                      rep('Shared',ns)), nt), rep('Shared',nt*2)) #2: variation, dependence
nonspat <- 0 #specify loading results for spatial or nonspatial model
mat3 <- readMat(paste0('./mat',nonspat,'.mat'))$matAll
P <- p*nzone + ns # pooled 3 ecozones plus 3 state effect
nrow(mat3) == (P*(nt+1) + P*(P+1)/2 + 2*nt) #both = 429

## [1] TRUE

# Dimension: Beta_t*(T+1) + size of Sigma_eta + spatial dependence and variation
p0 <- length(subvec <- c(c(1:(P*nt))+P, P*(nt+1) + P*(P+1)/2+(1:(2*nt)) ))
mat <- as.data.frame(cbind(mat3[subvec, c(1,3,4)], c(rep(rep(years,each=P),1),rep(years,2)),
                         varlabs0, uezonenew ))
for(j in 1:4) mat[,j] <- as.numeric(as.character(mat[,j]))
names(mat) <- c('means','lower','upper','year','vars','site')
mat$vars <- as.character(mat$vars)
mat$vars <- factor(mat$vars, levels=unique(varlabs0))
cols <- brewer.pal(3,'Set1') #c('red','green','blue','black')
pchs <- c(17,15,16,16)
lattice.options(default.theme = canonical.theme(color = FALSE))
my_padding <- list(strip = .9, top.padding = 0, main.key.padding = 0, key.axis.padding = 0,
                    axis.ylab.padding = 0, ylab.key.padding = 0, right.padding = 0, left.padding = 0,
                    key.sub.padding = 0, bottom.padding = 0, axis.right = 0, key.right = 0)
adjs <- c(-.2,0,.2,0) #jittering the x positions to distinguish site
plotBars <- function(mat, varList=NULL, layouts=c(3,3)){
  if(is.null(varList)) varList <- levels(mat$vars)
  nlab <- 3 #adhoc: both nzone=ns=3 here
  useDat <- droplevels(mat[mat$vars %in% varList, ])
  labs <- levels(useDat$site)[1:nlab]
  useDat$x <- useDat$year + adjs[as.integer(useDat$site)]
  fig <- xYplot(Cbind(means, lower, upper) ~ x|vars, groups=site, data=useDat,
                 ylab='', xlab='', pch=pchs, cex = .7, col=cols,
                 ylim = range(mat[,c('means','lower','upper')])+c(-1,1)*.05,
                 strip = strip.custom(par.strip.text = list(cex=.8, font=1)),
                 layout = layouts, abline=list(list(h=0,lty=2,col='black',lwd=1),list(v=1)),
                 as.table=T #index.cond = list(c(7,8,9,4,5,6,1,2,3))
  )
  fig <- update(fig, scales = list(alternating = F, cex = .6, abbreviate = F,
                                    y=list(relation = "free",cex=.6, tck=c(.4,0), rot = c(40,0)),
                                    x=list(at = years, cex=.6, tck = c(.3, 0), rot = c(40,0)) ),
                par.settings = list(layout.heights = my_padding, layout.widths = my_padding),
                key = list(text = list(labs), border = F, space='top', cex=.7, #corner = c(1,.8),
                           lines = list(lwd = 1,col = cols[1:nlab], lty=1),
                           points = list(cex = .8, pch=pchs[1:nlab],col = cols[1:nlab]))
  )
}
print(fig)
}

##### plot all effects

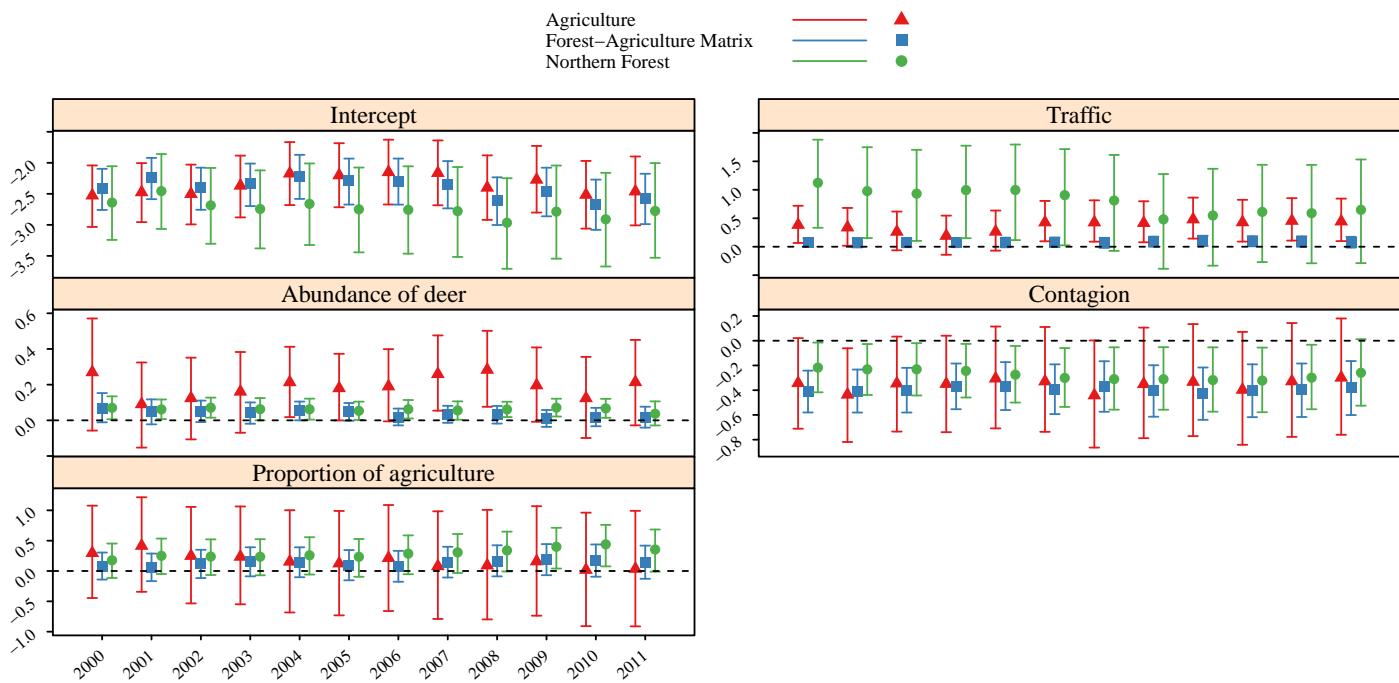
```

```

# if(saveFig)
#   postscript(paste0('./','beta_area_main_PPds',nonspat,'.eps'),
#             width = 6.4, height = 3.64, onefile=T, pointsize=5, horizontal=F,paper='special')
# plotBars(mat, varList=unique(varlabs0)[-1], layouts=c(3,3))
# if(saveFig) dev.off()

##### plot main effects
if(saveFig) pdf('beta_area_main.pdf', width = 8.2, height = 4, pointsize=16)
plotBars(mat, varList=varlabs1[1:p], layouts=c(2,3))

```



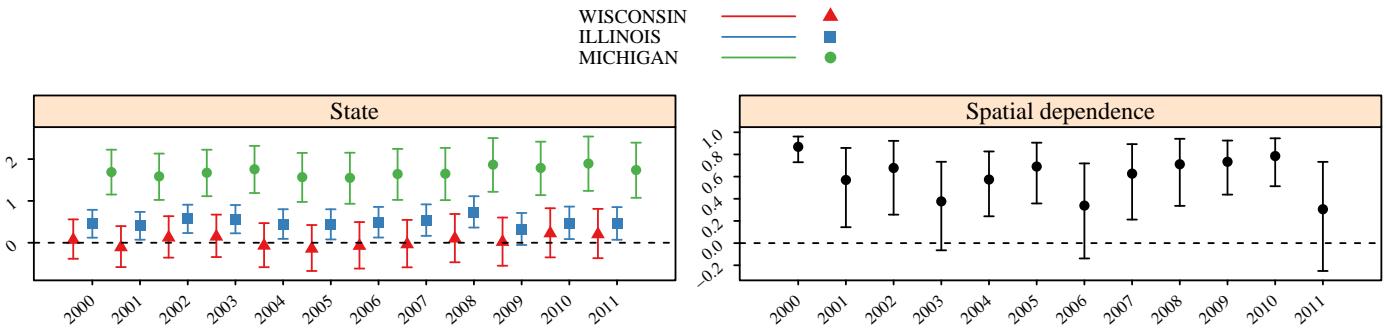
```

if(saveFig) dev.off()

useVar <- c(rev(unique(varlabs0))[c(5,4,3,1)])
# introduce state as grouping variable (site)
tmp <- droplevels(mat[mat$vars %in% useVar,])
pattern <- 'STATE: '
tmp$site <- factor(gsub(pattern, '', tmp$vars), levels=c('WI','IL','MI','spatial dependence'))
levels(tmp$site)[1:3] <- c('WISCONSIN','ILLINOIS','MICHIGAN')
inds <- which(grepl(pattern, tmp$vars))
tmp$vars <- as.character(tmp$vars); tmp$vars[inds] <- 'State'
tmp$vars <- factor(tmp$vars, levels=c('State','spatial dependence'))
levels(tmp$vars)[2] <- "Spatial dependence" #first letter capital
tmp$year <- tmp$year + adjs[as.integer(tmp$site)]

if(saveFig) pdf('beta_area_spat.pdf', width = 8.2, height = 2, pointsize=16)
plotBars(tmp, layouts=c(2,1))

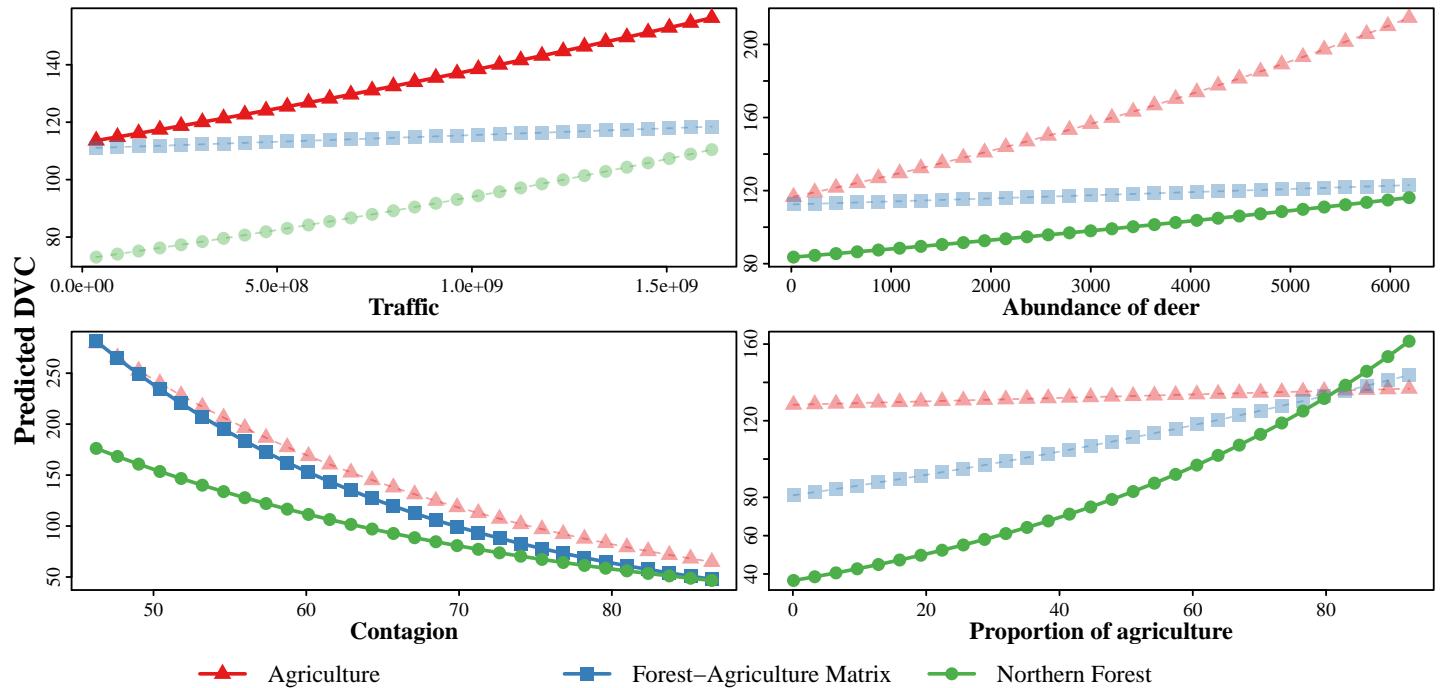
```



```
if(saveFig) dev.off()
```

Plot the biological effects for the influential variables on raw scale

```
#####
# plot predicted DVC versus influential variables, holding other constant
useYear <- 2010 #choose year
if(saveFig) pdf('single_effect.pdf', width = 8.2, height = 4, pointsize=14)
m0 <- matrix(c(1:4,5,5), nrow=3, byrow=TRUE) #use customized layout
layout(mat=m0, heights=c(.46, .46, .08)); par(oma=c(0,2,0,0), font.lab=1, cex.axis=1)
for(useVar in names(ss)[-1]){ #1=DVC
  inds <- which(years==useYear)
  summary(E[indT==inds]) #check range of offset
  meanE <- 1650; len <- 30
  s <- as.data.frame(t( ss[[useVar]][,inds] )) #summary statistics in raw scale
  x <- seq(s$min, s$max, len=len) #choose values span range of raw scale
  b0 <- mat[mat$vars=='Intercept' & mat$year==useYear, ]
  b1 <- mat[mat$vars==useVar & mat$year==useYear, ]
  b2 <- mat[mat$vars==paste0(pattern, 'MI')] & mat$year==useYear, ]
  y <- matrix(NA, len, nzone*2) #for 3 ecozones and 2 states (IA, MI)
  ltys <- lwds <- rep(NA, ncol(y))
  for(i in 1:nzone){
    j0 <- which(b0$site==uzone[i]); j1 <- which(b1$site==uzone[i])
    #holding other 0, note no state effect added, so this is for IA
    y[,i] <- b0$means[j0] + (x-s$mean)/s$stdev * b1$means[j1]
    y[,i+nzone] <- meanE*exp(y[,i] + b2$means) #for another state
    y[,i] <- meanE*exp(y[,i])
    isSignf <- b1$lower[j1]*b1$upper[j1]>0 #significance
    ltys[i] <- ifelse(isSignf, 1, 2); lwds[i] <- ifelse(isSignf, 2, 1)
  }
  par(mar=c(2.4,1,.4,.4), mgp=c(1.3,.3,0), tck=-.02, cex.lab=1.3, font.lab=2)
  matplot(x, y[,1:nzone], type='n', ylab='', xlab=useVar, family="Times")
  # grid(col='gray60')
  for(i in 1:nzone){
    useCol <- ifelse(ltys[i]==1, cols[i], adjustcolor(cols[i], alpha.f=.4))
    points(x,y[,i], col=useCol, pch=pchs[i], cex=1.5)
    lines(x,y[,i], col=useCol, lty=ltys[i], lwd=lwds[i])
  }
}
par(mar=c(0,0,0,0), family="Times")
plot(1, type="n", axes=FALSE, xlab="", ylab="")
lab <- uzone; lab[2] <- paste(lab[2], " ") #padding
legend('center', legend=lab, col=cols, pch=pchs, lwd=2, bty='n', horiz=T, cex=1.3, pt.cex=1.5)
par(las=0); mtext('Predicted DVC', side=2, line=.4, outer=TRUE, cex=1, font=2, family="Times")
```



```
if(saveFig) dev.off()
```