All\_SPP

Jesse Wood

September 13, 2018

## INDIVIDUAL SPECIES

setup for all

library("unmarked")

## Loading required package: reshape

## Loading required package: lattice

## Loading required package: parallel

## Loading required package: Rcpp

library("AICcmodavg")  
library("VGAM")

## Loading required package: stats4

##   
## Attaching package: 'stats4'

## The following object is masked from 'package:unmarked':  
##   
## mle

## Loading required package: splines

##   
## Attaching package: 'VGAM'

## The following object is masked from 'package:AICcmodavg':  
##   
## AICc

library("ggplot2")  
setwd("C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds")

# bhnu

bhnu.abund<- csvToUMF("bhnu\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(bhnu.abund)  
#str(bhnu.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(bhnu.abund)= scale (obsCovs(bhnu.abund))  
#select particular site covariates to scale below  
#(note: NOT ALL - not year, treatment, herbicide, last years ones)  
sc <- siteCovs(bhnu.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(bhnu.abund) <- sc

#test for NB or Poisson - most should use Poisson  
testP.bhnu <- pcount(~1 ~1, bhnu.abund, mixture="P", K=4)  
testNB.bhnu <- pcount(~1 ~1, bhnu.abund, mixture="NB", K=4)  
fmsTEST <- fitList(testP.bhnu, testNB.bhnu)

## Warning in fitList(testP.bhnu, testNB.bhnu): Your list was unnamed, so  
## model names were added as object names

msTEST.bhnu <- modSel(fmsTEST)  
msTEST.bhnu

## nPars AIC delta AICwt cumltvWt  
## testP.bhnu 2 587.30 0.00 0.73 0.73  
## testNB.bhnu 3 589.31 2.01 0.27 1.00

#Poisson is best for bhnu.

#test for some possible quadratic terms  
testR.bhnu <- pcount(~1 ~BA, bhnu.abund, mixture="P", K=4)  
testQ.bhnu <- pcount(~1 ~BA + I(BA^2), bhnu.abund, mixture="P", K=4)  
testC.bhnu <- pcount(~1 ~BA + I(BA^2) + I(BA^3), bhnu.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.bhnu, testQ.bhnu, testC.bhnu)

## Warning in fitList(testR.bhnu, testQ.bhnu, testC.bhnu): Your list was  
## unnamed, so model names were added as object names

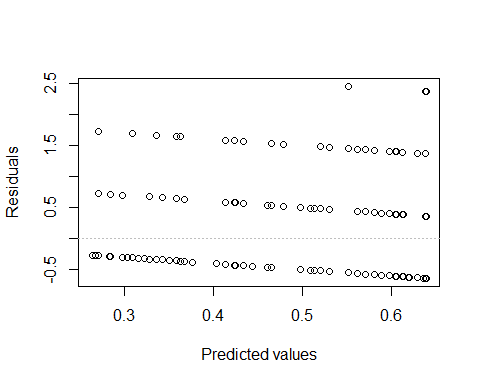
BAtest.bhnu <- modSel(msBAtest)  
BAtest.bhnu

## nPars AIC delta AICwt cumltvWt  
## testC.bhnu 5 586.46 0.00 0.37 0.37  
## testQ.bhnu 4 586.66 0.20 0.33 0.70  
## testR.bhnu 3 586.87 0.42 0.30 1.00

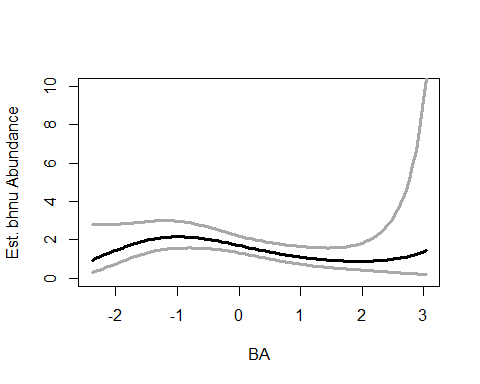
#  
testC.bhnu

##   
## Call:  
## pcount(formula = ~1 ~ BA + I(BA^2) + I(BA^3), data = bhnu.abund,   
## K = 4, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.5376 0.129 4.17 3.04e-05  
## BA -0.4072 0.176 -2.32 2.04e-02  
## I(BA^2) -0.1023 0.073 -1.40 1.61e-01  
## I(BA^3) 0.0714 0.044 1.62 1.05e-01  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## -0.869 0.156 -5.59 2.32e-08  
##   
## AIC: 586.4569

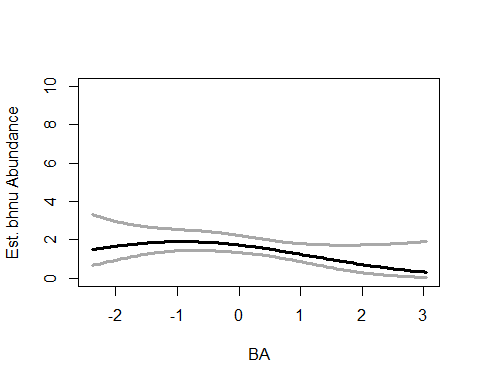
plot(testC.bhnu)



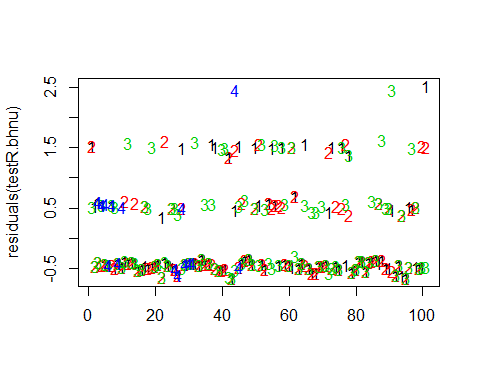
testC.bhnu <- pcount(~1 ~BA + I(BA^2) + I(BA^3), bhnu.abund, mixture="P", K=4)  
NewData.BA <-data.frame(BA=seq(min(sc$BA),max(sc$BA),length=100))  
bhnu.est.ba <- predict(testC.bhnu, type="state",  
 newdata=NewData.BA,appendData=TRUE)  
  
plot(Predicted~ BA, data=bhnu.est.ba, ylim=c(0,10), type="l", lwd=3,  
 xlab="BA", ylab="Est. bhnu Abundance")  
##95% confidence intervals  
lines(lower~ BA, data=bhnu.est.ba, type="l", lwd=3, col="darkgray")  
lines(upper~ BA, data=bhnu.est.ba, type="l", lwd=3, col="darkgray")



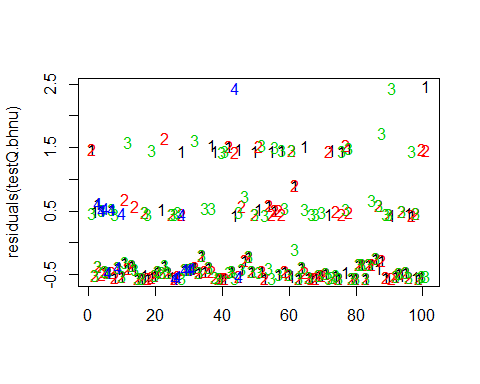
testQ.bhnu <- pcount(~1 ~BA + I(BA^2), bhnu.abund, mixture="P", K=4)  
NewData.BA <-data.frame(BA=seq(min(sc$BA),max(sc$BA),length=100))  
bhnu.est.ba <- predict(testQ.bhnu, type="state",  
 newdata=NewData.BA,appendData=TRUE)  
  
plot(Predicted~ BA, data=bhnu.est.ba, ylim=c(0,10), type="l", lwd=3,  
 xlab="BA", ylab="Est. bhnu Abundance")  
##95% confidence intervals  
lines(lower~ BA, data=bhnu.est.ba, type="l", lwd=3, col="darkgray")  
lines(upper~ BA, data=bhnu.est.ba, type="l", lwd=3, col="darkgray")



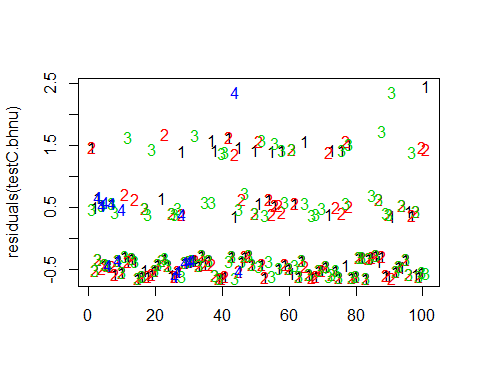
matplot(residuals(testR.bhnu))



matplot(residuals(testQ.bhnu))



matplot(residuals(testC.bhnu))



#test for some possible quadratic terms  
testRhw.bhnu <- pcount(~1 ~HW\_dens\_1050, bhnu.abund, mixture="P", K=4)  
testQhw.bhnu <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), bhnu.abund, mixture="P", K=4)  
testChw.bhnu <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2) + I(HW\_dens\_1050^3), bhnu.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.bhnu, testQ.bhnu, testC.bhnu)

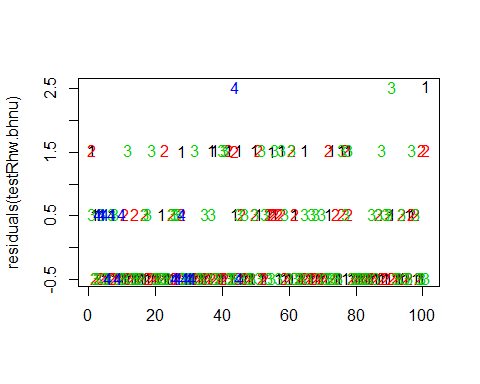
## Warning in fitList(testR.bhnu, testQ.bhnu, testC.bhnu): Your list was  
## unnamed, so model names were added as object names

HW1050test.bhnu <- modSel(msHW1050test)  
HW1050test.bhnu

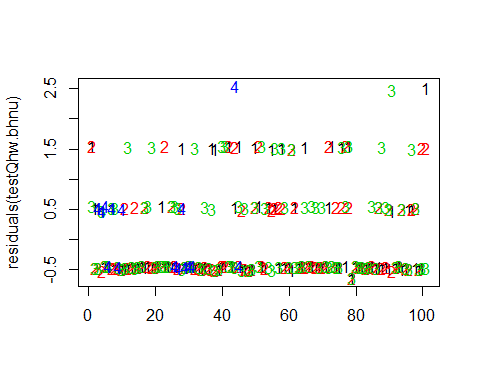
## nPars AIC delta AICwt cumltvWt  
## testC.bhnu 5 586.46 0.00 0.37 0.37  
## testQ.bhnu 4 586.66 0.20 0.33 0.70  
## testR.bhnu 3 586.87 0.42 0.30 1.00

#

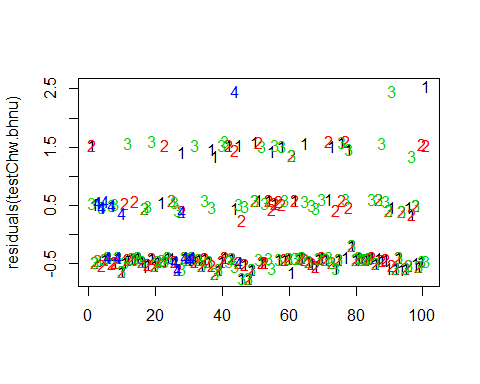
matplot(residuals(testRhw.bhnu))



matplot(residuals(testQhw.bhnu))



matplot(residuals(testChw.bhnu))



det.date.bhnu <- pcount(~ Jdate ~1, bhnu.abund, mixture="P", K=15)  
det.date2.bhnu <- pcount(~ Jdate + I(Jdate^2) ~1, bhnu.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.bhnu, det.date2.bhnu)

## Warning in fitList(det.date.bhnu, det.date2.bhnu): Your list was unnamed,  
## so model names were added as object names

DATEtest.bhnu <- modSel(mstestDATE)  
DATEtest.bhnu

## nPars AIC delta AICwt cumltvWt  
## det.date.bhnu 3 575.44 0.00 0.53 0.53  
## det.date2.bhnu 4 575.69 0.25 0.47 1.00

det.time.bhnu <-pcount(~ Time ~1, bhnu.abund, mixture="P",K=15)  
det.time2.bhnu <-pcount(~ Time + I(Time^2) ~1, bhnu.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.bhnu, det.time2.bhnu)

## Warning in fitList(det.time.bhnu, det.time2.bhnu): Your list was unnamed,  
## so model names were added as object names

TIMEtest.bhnu <- modSel(mstestTIME)  
TIMEtest.bhnu

## nPars AIC delta AICwt cumltvWt  
## det.time.bhnu 3 573.84 0.00 0.69 0.69  
## det.time2.bhnu 4 575.43 1.59 0.31 1.00

#detection covariates first  
det.null.bhnu <- pcount(~1 ~1, bhnu.abund, mixture="P", K=15)  
det.weather.bhnu <- pcount(~ Wind + Sky ~1, bhnu.abund, mixture="P", K=15)  
det.global.bhnu <- pcount(~ Jdate + Wind + Sky + Noise +Time ~1, bhnu.abund, mixture="P", K=15)  
det.sound.bhnu <- pcount(~ Noise + Wind ~1, bhnu.abund, mixture="P", K=15)  
det.date.bhnu <- pcount(~ Jdate ~1, bhnu.abund, mixture="P", K=15)  
det.detect.bhnu <- pcount(~ Jdate + Noise + Time ~1, bhnu.abund, mixture="P", K=15)  
det.notdate.bhnu <-pcount(~ Wind + Sky + Noise ~1, bhnu.abund, mixture="P", K=15)  
det.time.bhnu <-pcount(~ Time ~1, bhnu.abund, mixture="P",K=15)  
  
fmsDC <- fitList(det.null.bhnu, det.weather.bhnu, det.global.bhnu,  
 det.sound.bhnu, det.date.bhnu, det.detect.bhnu, det.notdate.bhnu, det.time.bhnu)

## Warning in fitList(det.null.bhnu, det.weather.bhnu, det.global.bhnu,  
## det.sound.bhnu, : Your list was unnamed, so model names were added as  
## object names

msDC.bhnu <- modSel(fmsDC)  
msDC.bhnu

## nPars AIC delta AICwt cumltvWt  
## det.detect.bhnu 5 570.66 0.00 0.5571 0.56  
## det.global.bhnu 7 572.73 2.07 0.1980 0.76  
## det.time.bhnu 3 573.84 3.18 0.1138 0.87  
## det.date.bhnu 3 575.44 4.78 0.0511 0.92  
## det.weather.bhnu 4 576.10 5.44 0.0367 0.96  
## det.null.bhnu 2 577.38 6.72 0.0194 0.98  
## det.notdate.bhnu 5 578.00 7.34 0.0142 0.99  
## det.sound.bhnu 4 578.76 8.10 0.0097 1.00

#msDC.bhnu@Full  
#summary: changed with 2018 data   
#2017 old: 1st is Time, 2nd null, 3rd detect (Jdate,Noise,Time), 4th date (Jdate)  
#Both yrs: det.detect.bhnu is top model and only one <d2. (global is 2.07)

det.detect.bhnu #+ relationship with Date & Time

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time ~ 1, data = bhnu.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.838 0.265 3.16 0.00157  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.3904 0.339 -4.103 4.08e-05  
## Jdate 0.2638 0.102 2.581 9.85e-03  
## Noise 0.0119 0.107 0.112 9.11e-01  
## Time 0.2995 0.105 2.866 4.16e-03  
##   
## AIC: 570.6627

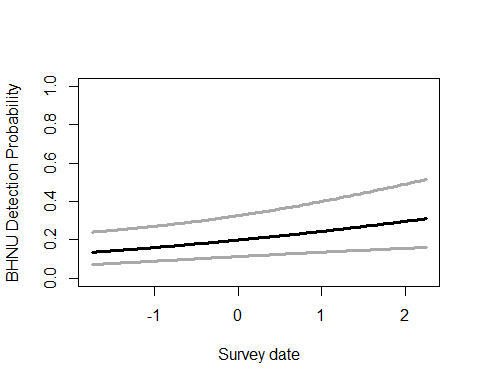
confint(det.detect.bhnu, type="det",method="normal")

## 0.025 0.975  
## p(Int) -2.05454131 -0.7262112  
## p(Jdate) 0.06349451 0.4641280  
## p(Noise) -0.19764907 0.2215470  
## p(Time) 0.09467010 0.5044128

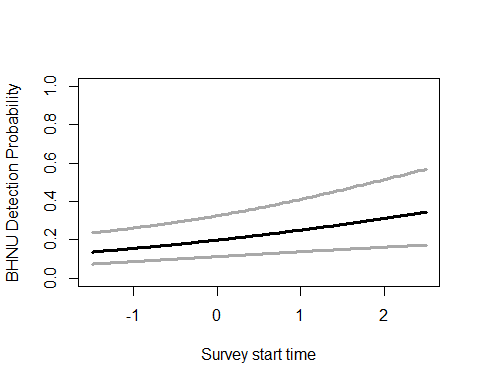
confint(det.detect.bhnu, type="state",method="normal") #significant

## 0.025 0.975  
## lam(Int) 0.3183654 1.356925

#date within detect model  
det.detect.bhnu <- pcount(~ Jdate + Noise + Time ~1, bhnu.abund, mixture="P", K=15)  
  
ND.bhnud <-data.frame(Jdate=seq(-1.75,2.25,length=100),Time=0,Noise=0)  
bhnu.est.date <- predict(det.detect.bhnu, type="det",  
 newdata=ND.bhnud,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=bhnu.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey date", ylab="BHNU Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=bhnu.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=bhnu.est.date, type="l", lwd=3, col="darkgray")



#time within detect model  
det.detect.bhnu <- pcount(~ Jdate + Noise + Time ~1, bhnu.abund, mixture="P", K=15)  
  
#summary(obsCovs(bhnu.abund))  
ND.bhnud2 <-data.frame(Time=seq(-1.5,2.5,length=100),Jdate=0,Noise=0)  
bhnu.est.time <- predict(det.detect.bhnu, type="det",  
 newdata=ND.bhnud2,appendData=TRUE)  
  
plot(Predicted~ Time, data=bhnu.est.time, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey start time", ylab="BHNU Detection Probability")  
##95% confidence intervals  
lines(lower~ Time, data=bhnu.est.time, type="l", lwd=3, col="darkgray")  
lines(upper~ Time, data=bhnu.est.time, type="l", lwd=3, col="darkgray")



write.table(msDC.bhnu@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/bhnu\_top\_models\_msDC.xls",sep="\t")

^bhnu: detection model best fit, + relationship with survey date & time

#run this when have CSV with both years  
null.bhnu <- pcount(~Jdate + Noise + Time ~1, bhnu.abund, mixture="P", K=40)  
year.bhnu <- pcount(~Jdate + Noise + Time ~ YearCat, bhnu.abund, mixture="P", K=40)  
fms.year.bhnu<- fitList(null.bhnu, year.bhnu)

## Warning in fitList(null.bhnu, year.bhnu): Your list was unnamed, so model  
## names were added as object names

year.ms.bhnu<-modSel(fms.year.bhnu)  
year.ms.bhnu

## nPars AIC delta AICwt cumltvWt  
## null.bhnu 5 570.66 0.00 0.65 0.65  
## year.bhnu 6 571.94 1.28 0.35 1.00

^ null better fit, but year model came in under d<2 (1.28) so… imp to include?

What you can & can’t use: \*as with before, just don’t use Water, or LowDev, or Impervious HWdens\_1050 & NHW\_saplings ok 500m: evergreen & ag ok, scrubs & ag ok 1km: scrubs & ag ok

500 # fine to use evergreen & ag # fine to use scrubs & ag 1 #### can’t use evergreen & ag at 1km # fine to use scrubs & ag 5 # - can’t use Evergreen&Ag, #+ can’t use HighDev&OpenDev together #### can’t use open & water #### can’t use evergreen & open #### can’t use Ag & scrubs 30 #### can’t use evergreen & protected together!! #### can’t use evergreen & scrubs together!! #### can’t use ag & water together!! #### can’t use open & scrubs together!! #+ can’t use Water&Protected together  
#+ can’t use Ag&OpenDev together #+ can’t use Grass&Ag together #- can’t use Protected&Ag together #- can’t use Ag&HighDev together #- can’t use HighDev&OpenDev together #- can’t use Evergreen&Ag together #- can’t use Schrubs&OpenDev together #+ fine to use Schrubs&HighDev together

updated detection covariates

#other detection covariates (Detection best model given msDC.bhnu)  
null.bhnu <- pcount(~ Jdate + Noise + Time ~1, bhnu.abund, mixture="P", K=40)  
global.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + Herbicide + BA + Nsnags +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + NP\_over\_20cm + Rel\_HW2P\_canopy   
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , bhnu.abund, mixture="P", K=40) #FPSiteIndex removed  
local.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , bhnu.abund, mixture="P", K=40) #can only include BA OR CCover  
lh.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Ccover + Age + Nsnags + TreeHt  
 + NP\_over\_20cm + Rel\_HW2P\_canopy + YearCat  
 , bhnu.abund, mixture="P", K=40)  
landmetrics.bhnu <- pcount (~ Jdate + Noise + Time  
 ~ Parea + ShapeIndex + YearCat  
 , bhnu.abund, mixture="P",K=40)  
landscape500.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen500m + HighDev500m + OpenDev500m + YearCat  
 , bhnu.abund, mixture="P", K=40)  
landscape1.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen1km + HighDev1km + OpenDev1km + YearCat  
 , bhnu.abund, mixture="P", K=40)  
landscape5.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen5km + OpenDev5km + YearCat  
 , bhnu.abund, mixture="P", K=40)  
landscape30.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen30km + OpenDev30km + YearCat  
 , bhnu.abund, mixture="P", K=40) #rmoved Protected  
treatment.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + Nthins + YearCat  
 , bhnu.abund, mixture ="P", K=40)  
management.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + BA + TimeSinceB + TimeSinceT  
 + Herbicide + YearCat  
 , bhnu.abund, mixture="P", K=40)  
management2.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + BA + I(BA^2) + TimeSinceB + TimeSinceT  
 + Herbicide + YearCat  
 , bhnu.abund, mixture="P", K=40)  
#management3.bhnu <- pcount(~ Jdate + Noise + Time  
# ~ Treatment + BA + I(BA^2) + I(BA^3) + TimeSinceB + TimeSinceT  
# + Herbicide + YearCat  
# , bhnu.abund, mixture="P", K=40)  
disturbance.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , bhnu.abund, mixture="P", K=40)  
siteprod.bhnu <- pcount(~ Jdate + Noise + Time ~ PISoils + NSoilTypes + YearCat  
 , bhnu.abund, mixture="P", K=40) #FPSiteIndex removed  
upstate.bhnu <- pcount(~ Jdate + Noise + Time ~ Parea + HighDev5km + BA  
 + TreeHt + YearCat  
 , bhnu.abund, mixture="P", K=40)  
upstate2.bhnu <- pcount(~ Jdate + Noise + Time ~ Parea + HighDev5km + BA + I(BA^2)  
 + TreeHt + YearCat  
 , bhnu.abund, mixture="P", K=40)  
coord.bhnu <- pcount (~Jdate + Noise + Time ~ Latitude + Longitude + YearCat  
 , bhnu.abund, mixture="P", K=80)  
  
  
fms <- fitList(null.bhnu, global.bhnu, local.bhnu, lh.bhnu, landmetrics.bhnu,  
 landscape500.bhnu, landscape1.bhnu, landscape5.bhnu, landscape30.bhnu,treatment.bhnu, management.bhnu, management2.bhnu, disturbance.bhnu, siteprod.bhnu, upstate.bhnu, upstate2.bhnu, coord.bhnu)

## Warning in fitList(null.bhnu, global.bhnu, local.bhnu, lh.bhnu,  
## landmetrics.bhnu, : Your list was unnamed, so model names were added as  
## object names

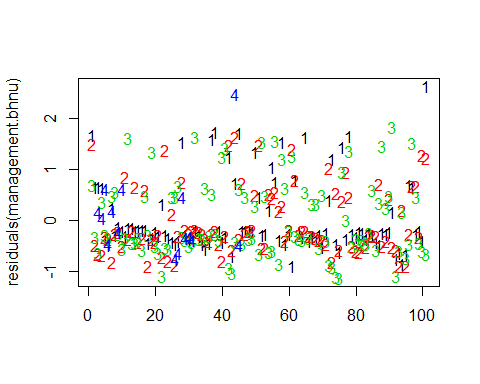
ms.bhnu <- modSel(fms) #note this does not include some of site.prod, nor management3.bhnu,  
ms.bhnu

## nPars AIC delta AICwt cumltvWt  
## treatment.bhnu 10 565.16 0.00 3.5e-01 0.35  
## lh.bhnu 12 565.61 0.45 2.8e-01 0.63  
## landscape1.bhnu 9 566.82 1.66 1.5e-01 0.78  
## management2.bhnu 14 569.38 4.22 4.3e-02 0.83  
## disturbance.bhnu 8 570.42 5.26 2.5e-02 0.85  
## upstate2.bhnu 11 570.48 5.31 2.5e-02 0.88  
## local.bhnu 9 570.57 5.41 2.3e-02 0.90  
## null.bhnu 5 570.66 5.50 2.2e-02 0.92  
## management.bhnu 13 570.94 5.78 1.9e-02 0.94  
## coord.bhnu 8 571.25 6.08 1.7e-02 0.96  
## landscape500.bhnu 9 571.25 6.09 1.7e-02 0.98  
## upstate.bhnu 10 571.53 6.37 1.5e-02 0.99  
## siteprod.bhnu 8 574.25 9.08 3.7e-03 0.99  
## landmetrics.bhnu 8 574.94 9.78 2.6e-03 1.00  
## landscape5.bhnu 8 575.15 9.99 2.4e-03 1.00  
## landscape30.bhnu 8 575.87 10.70 1.7e-03 1.00  
## global.bhnu 25 582.95 17.79 4.8e-05 1.00

#ms.bhnu@Full

changed with 2018 data - now Tx is best and LH is second best, landscape 1 third best #changed AGAIN with cubing of BA - now cubic BA management model is 4th best, also under delta 2.0! but cubic doesn’t make sense - when ran with quadratic, that model 4th best but >2.0

matplot(residuals(management.bhnu))



treatment.bhnu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time ~ Treatment + Nthins +   
## YearCat, data = bhnu.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.4689 0.422 1.112 0.26605  
## Treatment1B 0.3055 0.337 0.907 0.36433  
## Treatment2B 0.3931 0.333 1.182 0.23732  
## Treatment3B 1.0828 0.340 3.184 0.00145  
## Nthins -0.1049 0.117 -0.894 0.37139  
## YearCatB 0.0886 0.189 0.468 0.64007  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.6975 0.4305 -3.943 8.06e-05  
## Jdate 0.2590 0.0993 2.609 9.08e-03  
## Noise 0.0499 0.1035 0.483 6.29e-01  
## Time 0.2929 0.1006 2.910 3.61e-03  
##   
## AIC: 565.1624

confint(treatment.bhnu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.3574214 1.2952507  
## lam(Treatment1B) -0.3545929 0.9656418  
## lam(Treatment2B) -0.2588832 1.0450492  
## lam(Treatment3B) 0.4163208 1.7492843  
## lam(Nthins) -0.3347948 0.1250666  
## lam(YearCatB) -0.2826156 0.4597194

lh.bhnu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time ~ Ccover + Age + Nsnags +   
## TreeHt + NP\_over\_20cm + Rel\_HW2P\_canopy + YearCat, data = bhnu.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.9918 0.4159 2.385 0.01710  
## Ccover -0.0319 0.1037 -0.308 0.75846  
## Age 0.1297 0.1041 1.245 0.21299  
## Nsnags 0.0516 0.0871 0.593 0.55314  
## TreeHt 0.0491 0.1151 0.427 0.66965  
## NP\_over\_20cm -0.0777 0.1160 -0.670 0.50279  
## Rel\_HW2P\_canopy -0.3693 0.1166 -3.168 0.00153  
## YearCatB 0.2147 0.2132 1.007 0.31400  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.8148 0.4925 -3.685 0.000229  
## Jdate 0.2441 0.0983 2.484 0.012983  
## Noise -0.0116 0.1019 -0.114 0.909062  
## Time 0.3073 0.1015 3.029 0.002455  
##   
## AIC: 565.6129

confint(lh.bhnu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.17662367 1.8069652  
## lam(Ccover) -0.23521186 0.1714155  
## lam(Age) -0.07441479 0.3337987  
## lam(Nsnags) -0.11899473 0.2222509  
## lam(TreeHt) -0.17646267 0.2746573  
## lam(NP\_over\_20cm) -0.30502615 0.1495940  
## lam(Rel\_HW2P\_canopy) -0.59780365 -0.1408651  
## lam(YearCatB) -0.20319934 0.6325280

landscape1.bhnu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time ~ Evergreen1km + HighDev1km +   
## OpenDev1km + YearCat, data = bhnu.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.850 0.3181 2.672 0.00755  
## Evergreen1km 0.215 0.0929 2.312 0.02079  
## HighDev1km 0.157 0.0629 2.489 0.01279  
## OpenDev1km 0.119 0.1039 1.143 0.25324  
## YearCatB 0.159 0.1894 0.839 0.40160  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.56004 0.385 -4.0473 5.18e-05  
## Jdate 0.26683 0.101 2.6289 8.57e-03  
## Noise -0.00899 0.109 -0.0828 9.34e-01  
## Time 0.31866 0.104 3.0603 2.21e-03  
##   
## AIC: 566.8243

confint(landscape1.bhnu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.22639205 1.4732153  
## lam(Evergreen1km) 0.03267600 0.3966979  
## lam(HighDev1km) 0.03333305 0.2800887  
## lam(OpenDev1km) -0.08497141 0.3224971  
## lam(YearCatB) -0.21237716 0.5301348

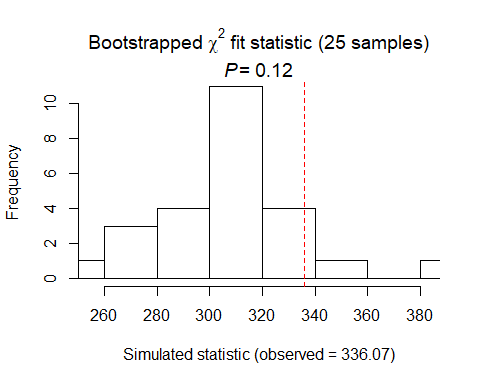
#management2.bhnu #just curious - this is no longer a top model  
#confint(management2.bhnu, type="state",method="normal")

bhnu treatment: only Treatment3B sig (+) bhnu life history: only Rel\_HW2P\_canopy is sig (-) bhnu landscape1: Evergreen1km(+), HighDev(+) — not top bhnu management2: BA^2(-) is close but not sig

#for figures: treatment model (Treatment), life history model (Rel\_HW2P\_canopy), landscape 1 model (evergreen1km, highdev1km)  
  
treatment.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + Nthins + YearCat  
 , bhnu.abund, mixture ="P", K=40)  
lh.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Ccover + Age + Nsnags + TreeHt  
 + NP\_over\_20cm + Rel\_HW2P\_canopy + YearCat  
 , bhnu.abund, mixture="P", K=40)  
landscape1.bhnu <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen1km + HighDev1km + OpenDev1km + YearCat  
 , bhnu.abund, mixture="P", K=40)

write.table(ms.bhnu@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/bhnu\_top\_models\_ms.xls",sep="\t")

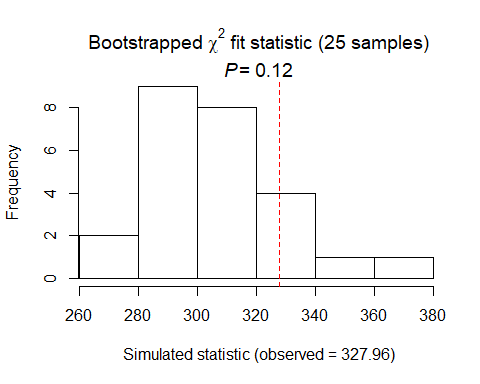
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 336.0727   
## Number of bootstrap samples = 25  
## P-value = 0.12  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 255 292 306 316 382   
##   
## Estimate of c-hat = 1.09

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 336.0727

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 327.9556   
## Number of bootstrap samples = 25  
## P-value = 0.12  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 264 288 302 318 378   
##   
## Estimate of c-hat = 1.07

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 327.9556

bhnu summary: P distribution DCs: detect best model (+ date, + time, non-sig with noise) SCs, using detect model: bhnu treatment: only Treatment3B sig (+) bhnu life history: only Rel\_HW2P\_canopy is sig (-) bhnu landscape1: Evergreen1km(+), HighDev(+)

# cawr

cawr.abund<- csvToUMF("cawr\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(cawr.abund)  
#str(cawr.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(cawr.abund)= scale (obsCovs(cawr.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(cawr.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(cawr.abund) <- sc

#run this when have CSV with both years  
null.cawr <- pcount(~Jdate + Noise + Time ~1, cawr.abund, mixture="P", K=40)  
year.cawr <- pcount(~Jdate + Noise + Time ~ YearCat, cawr.abund, mixture="P", K=40)  
fms.year.cawr<- fitList(null.cawr, year.cawr)

## Warning in fitList(null.cawr, year.cawr): Your list was unnamed, so model  
## names were added as object names

year.ms.cawr<-modSel(fms.year.cawr)  
year.ms.cawr

## nPars AIC delta AICwt cumltvWt  
## null.cawr 5 716.52 0.00 0.71 0.71  
## year.cawr 6 718.33 1.81 0.29 1.00

^null is best but year <d1.81

det.date.cawr <- pcount(~ Jdate ~1, cawr.abund, mixture="P", K=15)  
det.date2.cawr <- pcount(~ Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.cawr, det.date2.cawr)

## Warning in fitList(det.date.cawr, det.date2.cawr): Your list was unnamed,  
## so model names were added as object names

DATEtest.cawr <- modSel(mstestDATE)  
DATEtest.cawr

## nPars AIC delta AICwt cumltvWt  
## det.date2.cawr 4 710.53 0.00 0.91 0.91  
## det.date.cawr 3 715.17 4.64 0.09 1.00

#date quadratic  
  
det.time.cawr <-pcount(~ Time ~1, cawr.abund, mixture="P",K=15)  
det.time2.cawr <-pcount(~ Time + I(Time^2) ~1, cawr.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.cawr, det.time2.cawr)

## Warning in fitList(det.time.cawr, det.time2.cawr): Your list was unnamed,  
## so model names were added as object names

TIMEtest.cawr <- modSel(mstestTIME)  
TIMEtest.cawr

## nPars AIC delta AICwt cumltvWt  
## det.time.cawr 3 715.60 0.00 0.73 0.73  
## det.time2.cawr 4 717.58 1.99 0.27 1.00

#time not

#detection covariates first  
det.null.cawr <- pcount(~1 ~1, cawr.abund, mixture="P", K=15)  
det.weather.cawr <- pcount(~ Wind + Sky ~1, cawr.abund, mixture="P", K=15)  
det.global.cawr <- pcount(~ Jdate + Wind + Sky + Noise + Time ~1, cawr.abund, mixture="P", K=15)  
det.sound.cawr <- pcount(~ Noise + Wind ~1, cawr.abund, mixture="P", K=15)  
det.date.cawr <- pcount(~ Jdate ~1, cawr.abund, mixture="P", K=15)  
det.date2.cawr <- pcount(~ Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
det.detect.cawr <- pcount(~ Noise + Time + Jdate ~1, cawr.abund, mixture="P", K=15)  
det.detect2.cawr <- pcount(~ Noise + Time + Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
det.notdate.cawr <-pcount(~ Wind + Sky + Noise ~1, cawr.abund, mixture="P", K=15)  
det.time.cawr <-pcount(~ Time ~1, cawr.abund, mixture="P",K=15)  
det.timing.cawr <-pcount(~ Time + Jdate ~1, cawr.abund, mixture="P", K=15)  
det.timing2.cawr <-pcount(~ Time + Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.cawr, det.weather.cawr, det.global.cawr,  
 det.sound.cawr, det.date.cawr, det.date2.cawr,  
 det.detect.cawr, det.detect2.cawr, det.notdate.cawr,  
 det.time.cawr, det.timing.cawr, det.timing2.cawr)

## Warning in fitList(det.null.cawr, det.weather.cawr, det.global.cawr,  
## det.sound.cawr, : Your list was unnamed, so model names were added as  
## object names

msDC.cawr <- modSel(fmsDC)  
msDC.cawr

## nPars AIC delta AICwt cumltvWt  
## det.date2.cawr 4 710.53 0.00 0.3836 0.38  
## det.detect2.cawr 6 711.83 1.30 0.2002 0.58  
## det.timing2.cawr 5 711.91 1.38 0.1924 0.78  
## det.null.cawr 2 713.85 3.33 0.0727 0.85  
## det.date.cawr 3 715.17 4.64 0.0377 0.89  
## det.time.cawr 3 715.60 5.07 0.0304 0.92  
## det.sound.cawr 4 716.21 5.68 0.0224 0.94  
## det.detect.cawr 5 716.52 5.99 0.0191 0.96  
## det.timing.cawr 4 716.62 6.09 0.0183 0.98  
## det.weather.cawr 4 717.65 7.13 0.0109 0.99  
## det.notdate.cawr 5 718.13 7.60 0.0086 1.00  
## det.global.cawr 7 719.81 9.29 0.0037 1.00

#msDC.cawr@Full  
#summary: null best, date second best 0.21, detect (Jdate+Noise+Time) 1.57, then sound, time  
#new summary w quadratic - date now best, detect second, timing third! all with jdate^2

date2 model (+), then detect2 (date^2 +, rest nonsig), then timing2 (date^2 +, rest nonsig) (all under 2.0)

det.date2.cawr

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ 1, data = cawr.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.01 0.258 3.91 9.34e-05  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9770 0.3636 -2.69 0.00721  
## Jdate -0.0974 0.0752 -1.29 0.19555  
## I(Jdate^2) 0.1889 0.0758 2.49 0.01266  
##   
## AIC: 710.5267

confint(det.date2.cawr, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.68969212 -0.26435274  
## p(Jdate) -0.24485808 0.05008316  
## p(I(Jdate^2)) 0.04040828 0.33745442

confint(det.date2.cawr, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.5020027 1.512584

det.detect2.cawr

##   
## Call:  
## pcount(formula = ~Noise + Time + Jdate + I(Jdate^2) ~ 1, data = cawr.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.976 0.243 4.02 5.81e-05  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9373 0.3464 -2.706 0.00681  
## Noise 0.1313 0.0931 1.410 0.15853  
## Time -0.0737 0.0807 -0.913 0.36112  
## Jdate -0.1290 0.0801 -1.609 0.10752  
## I(Jdate^2) 0.1918 0.0769 2.496 0.01258  
##   
## AIC: 711.8277

confint(det.detect2.cawr, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.61616289 -0.25845145  
## p(Noise) -0.05122533 0.31390604  
## p(Time) -0.23185527 0.08446727  
## p(Jdate) -0.28607659 0.02809386  
## p(I(Jdate^2)) 0.04117337 0.34250850

confint(det.detect2.cawr, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.4999808 1.451118

det.timing2.cawr

##   
## Call:  
## pcount(formula = ~Time + Jdate + I(Jdate^2) ~ 1, data = cawr.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.01 0.256 3.92 8.72e-05  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9771 0.3619 -2.700 0.00693  
## Time -0.0623 0.0792 -0.787 0.43154  
## Jdate -0.1131 0.0781 -1.448 0.14775  
## I(Jdate^2) 0.1899 0.0758 2.506 0.01220  
##   
## AIC: 711.9072

confint(det.timing2.cawr, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.68632422 -0.26781018  
## p(Time) -0.21745096 0.09290179  
## p(Jdate) -0.26626026 0.04004288  
## p(I(Jdate^2)) 0.04138973 0.33836960

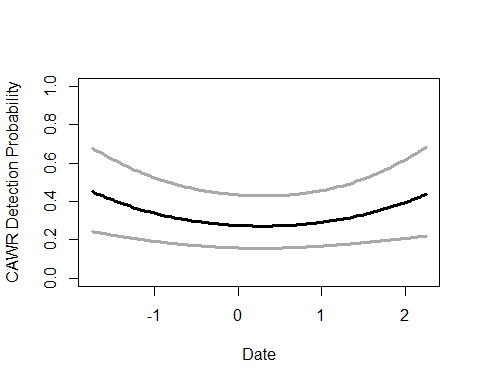
confint(det.timing2.cawr, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.5034384 1.508402

#date first - model-averaged  
det.date2.cawr <- pcount(~ Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
det.detect2.cawr <- pcount(~ Noise + Time + Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
det.timing2.cawr <-pcount(~ Time + Jdate + I(Jdate^2) ~1, cawr.abund, mixture="P", K=15)  
  
dms\_top.cawrd <- fitList(det.date2.cawr,det.detect2.cawr,det.timing2.cawr)

## Warning in fitList(det.date2.cawr, det.detect2.cawr, det.timing2.cawr):  
## Your list was unnamed, so model names were added as object names

ND.cawrd <-data.frame(Jdate=seq(-1.75,2.25,length=100),Noise=0,Time=0)  
cawr.est.date <- predict(dms\_top.cawrd, type="det",  
 newdata=ND.cawrd,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=cawr.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Date", ylab="CAWR Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=cawr.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=cawr.est.date, type="l", lwd=3, col="darkgray")



write.table(msDC.cawr@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/cawr\_top\_models\_msDC.xls",sep="\t")

#test for some possible quadratic terms - for BA, definitely NON quadratic  
testR.cawr <- pcount(~1 ~BA, cawr.abund, mixture="P", K=4)  
testQ.cawr <- pcount(~1 ~BA + I(BA^2), cawr.abund, mixture="P", K=4)  
testC.cawr <- pcount(~1 ~BA + I(BA^2) + I(BA^3), cawr.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.cawr, testQ.cawr, testC.cawr)

## Warning in fitList(testR.cawr, testQ.cawr, testC.cawr): Your list was  
## unnamed, so model names were added as object names

BAtest.cawr <- modSel(msBAtest)  
BAtest.cawr

## nPars AIC delta AICwt cumltvWt  
## testR.cawr 3 722.58 0.00 0.52 0.52  
## testQ.cawr 4 723.57 0.99 0.31 0.83  
## testC.cawr 5 724.80 2.22 0.17 1.00

#

# quadratic not better for cawr: BA, HW\_dens\_1050, FG\_herb, FG\_shrub, NHW\_saplings

testR.cawr <- pcount(~1 ~BA, cawr.abund, mixture="P", K=4)  
testQ.cawr <- pcount(~1 ~BA + I(BA^2), cawr.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.cawr, testQ.cawr)

## Warning in fitList(testR.cawr, testQ.cawr): Your list was unnamed, so model  
## names were added as object names

BAtest.cawr <- modSel(msBAtest)  
BAtest.cawr

## nPars AIC delta AICwt cumltvWt  
## testR.cawr 3 722.58 0.00 0.62 0.62  
## testQ.cawr 4 723.57 0.99 0.38 1.00

testR.cawr <- pcount(~1 ~HW\_dens\_1050, cawr.abund, mixture="P", K=4)  
testQ.cawr <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), cawr.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.cawr, testQ.cawr)

## Warning in fitList(testR.cawr, testQ.cawr): Your list was unnamed, so model  
## names were added as object names

HW1050test.cawr <- modSel(msHW1050test)  
HW1050test.cawr

## nPars AIC delta AICwt cumltvWt  
## testR.cawr 3 722.46 0.00 0.72 0.72  
## testQ.cawr 4 724.32 1.87 0.28 1.00

testR.cawr <- pcount(~1 ~FG\_herb, cawr.abund, mixture="P", K=4)  
testQ.cawr <- pcount(~1 ~FG\_herb + I(FG\_herb^2), cawr.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.cawr, testQ.cawr)

## Warning in fitList(testR.cawr, testQ.cawr): Your list was unnamed, so model  
## names were added as object names

FGHtest.cawr <- modSel(msFGHtest)  
FGHtest.cawr

## nPars AIC delta AICwt cumltvWt  
## testR.cawr 3 721.65 0.00 0.73 0.73  
## testQ.cawr 4 723.65 2.00 0.27 1.00

testR.cawr <- pcount(~1 ~FG\_shrub, cawr.abund, mixture="P", K=4)  
testQ.cawr <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), cawr.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.cawr, testQ.cawr)

## Warning in fitList(testR.cawr, testQ.cawr): Your list was unnamed, so model  
## names were added as object names

FGStest.cawr <- modSel(msFGStest)  
FGStest.cawr

## nPars AIC delta AICwt cumltvWt  
## testR.cawr 3 722.48 0.00 0.69 0.69  
## testQ.cawr 4 724.04 1.55 0.31 1.00

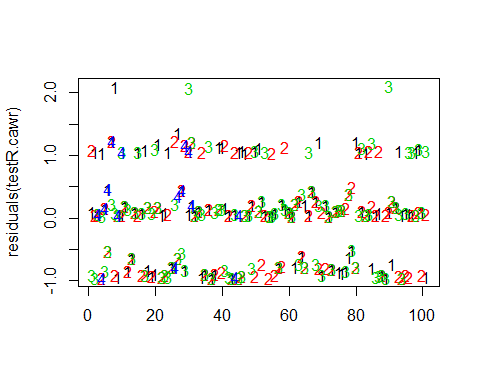
testR.cawr <- pcount(~1 ~NHW\_saplings, cawr.abund, mixture="P", K=4)  
testQ.cawr <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), cawr.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.cawr, testQ.cawr)

## Warning in fitList(testR.cawr, testQ.cawr): Your list was unnamed, so model  
## names were added as object names

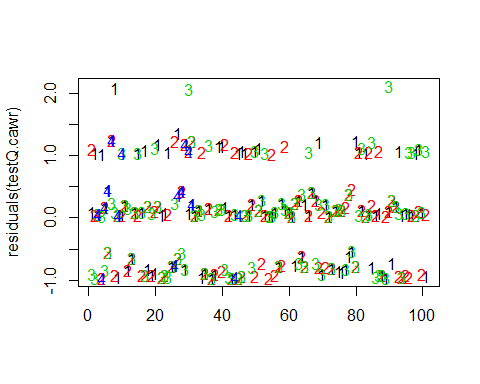
HWStest.cawr <- modSel(msHWStest)  
HWStest.cawr

## nPars AIC delta AICwt cumltvWt  
## testR.cawr 3 720.32 0.00 0.73 0.73  
## testQ.cawr 4 722.30 1.98 0.27 1.00

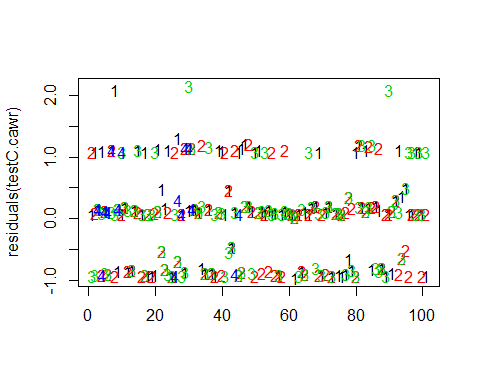
matplot(residuals(testR.cawr))



matplot(residuals(testQ.cawr))



matplot(residuals(testC.cawr))



##site covariates next  
#null detection covariates (this was the best model...run)  
#cawr (open woodlands, insects, cavity, foliage gleaner - brushy thickets, shrubby)   
# covariates: density (BA not Ccover), Nsnags, midstory (NHW\_saplings, midstory), OpenDev  
  
null.cawr <- pcount(~Jdate + I(Jdate^2) ~1  
 ,cawr.abund, mixture="P", K=40)  
global.cawr <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + Herbicide + BA + Nsnags +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + FG\_herb + FG\_shrub + NHW\_saplings  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , cawr.abund, mixture="P", K=40) #FPSiteIndex removed + Rel\_HW2P\_canopy + NP\_over\_20cm   
local.cawr <- pcount(~ Jdate + I(Jdate^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , cawr.abund, mixture="P", K=40)  
lh.cawr <- pcount(~ Jdate + I(Jdate^2)  
 ~ BA + Nsnags + HW\_dens\_1050 + FG\_herb + FG\_shrub + NHW\_saplings + YearCat  
 , cawr.abund, mixture="P", K=40)  
landmetrics.cawr <- pcount (~ Jdate + I(Jdate^2) ~ Parea + ShapeIndex + YearCat  
 , cawr.abund, mixture="P",K=40)  
landscape500.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Evergreen500m + OpenDev500m + HighDev500m + Schrubs500m + YearCat  
 , cawr.abund, mixture="P", K=40)  
landscape1.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Evergreen1km + OpenDev1km + HighDev1km + Schrubs1km + YearCat  
 , cawr.abund, mixture="P", K=40)  
landscape5.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Evergreen5km + HighDev5km + Schrubs5km + YearCat  
 , cawr.abund, mixture="P", K=40)  
landscape30.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Evergreen30km + OpenDev30km + YearCat  
 , cawr.abund, mixture="P", K=40)  
treatment.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Treatment + Nthins + YearCat  
 , cawr.abund, mixture ="P", K=40)  
management.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , cawr.abund, mixture="P", K=40)  
disturbance.cawr <- pcount(~ Jdate + I(Jdate^2) ~ TimeSinceB + TimeSinceT + YearCat  
 , cawr.abund, mixture="P", K=40)  
siteprod.cawr <- pcount(~ Jdate + I(Jdate^2) ~ PISoils + NSoilTypes + YearCat  
 , cawr.abund, mixture="P", K=40) #FPSiteIndex removed  
upstate.cawr <- pcount(~ Jdate + I(Jdate^2) ~ Ag5km + Parea + YearCat, cawr.abund, mixture="P", K=40)  
coord.cawr <- pcount (~Jdate + I(Jdate^2) ~ Latitude + Longitude + YearCat  
 , cawr.abund, mixture="P", K=40)  
  
fms <- fitList(null.cawr, global.cawr, local.cawr, lh.cawr, landmetrics.cawr, landscape500.cawr, landscape1.cawr, landscape5.cawr, landscape30.cawr,  
 treatment.cawr, management.cawr, disturbance.cawr,  
 siteprod.cawr, upstate.cawr, coord.cawr)

## Warning in fitList(null.cawr, global.cawr, local.cawr, lh.cawr,  
## landmetrics.cawr, : Your list was unnamed, so model names were added as  
## object names

ms.cawr <- modSel(fms)  
#ms.cawr@Full  
ms.cawr

## nPars AIC delta AICwt cumltvWt  
## null.cawr 4 710.53 0.00 3.7e-01 0.37  
## upstate.cawr 7 711.15 0.63 2.7e-01 0.64  
## landscape5.cawr 8 713.41 2.88 8.7e-02 0.72  
## landscape30.cawr 7 713.92 3.39 6.8e-02 0.79  
## landscape1.cawr 9 714.06 3.53 6.3e-02 0.85  
## landmetrics.cawr 7 715.07 4.54 3.8e-02 0.89  
## siteprod.cawr 7 715.74 5.22 2.7e-02 0.92  
## disturbance.cawr 7 716.08 5.56 2.3e-02 0.94  
## coord.cawr 7 716.36 5.83 2.0e-02 0.96  
## landscape500.cawr 9 716.94 6.41 1.5e-02 0.98  
## treatment.cawr 9 718.33 7.80 7.4e-03 0.99  
## local.cawr 8 718.34 7.81 7.4e-03 0.99  
## lh.cawr 11 718.48 7.95 6.9e-03 1.00  
## management.cawr 12 723.63 13.10 5.3e-04 1.00  
## global.cawr 26 742.36 31.83 4.5e-08 1.00

# summary: null is best model but upstate second best model (+ with Ag5km)  
upstate.cawr

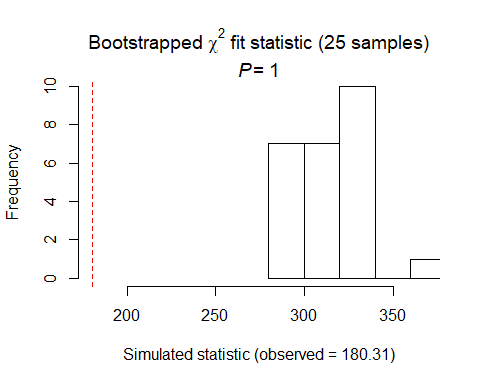
##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ Ag5km + Parea + YearCat,   
## data = cawr.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.065071 0.3143 3.388572 0.000703  
## Ag5km 0.151264 0.0724 2.090262 0.036594  
## Parea -0.072025 0.0832 -0.865395 0.386822  
## YearCatB 0.000154 0.1576 0.000979 0.999219  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.0684 0.4260 -2.51 0.0121  
## Jdate -0.0883 0.0740 -1.19 0.2331  
## I(Jdate^2) 0.1740 0.0758 2.29 0.0218  
##   
## AIC: 711.1528

confint(upstate.cawr, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.449029631 1.68111282  
## lam(Ag5km) 0.009429148 0.29309823  
## lam(Parea) -0.235148993 0.09109877  
## lam(YearCatB) -0.308733417 0.30904207

write.table(ms.cawr@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/cawr\_top\_models\_ms.xls",sep="\t")

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 180.3074   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 283 299 313 324 369   
##   
## Estimate of c-hat = 0.57

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 180.3074

cawr summary: P distribution DCs: DATE2 model, then DETECT2, then TIMING2 (all under 2.0) - + quad with date Using date^2 model for detection covariates: null model best upstate model second best (+ with ag5km)

# CACH

cach.abund<- csvToUMF("cach\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(cach.abund)  
#str(cach.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(cach.abund)= scale (obsCovs(cach.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(cach.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(cach.abund) <- sc

#run this when have CSV with both years  
null.cach <- pcount(~Jdate + Noise + Time ~1, cach.abund, mixture="P", K=40)  
year.cach <- pcount(~Jdate + Noise + Time ~ YearCat, cach.abund, mixture="P", K=40)  
fms.year.cach<- fitList(null.cach, year.cach)

## Warning in fitList(null.cach, year.cach): Your list was unnamed, so model  
## names were added as object names

year.ms.cach<-modSel(fms.year.cach)  
year.ms.cach

## nPars AIC delta AICwt cumltvWt  
## year.cach 6 786.85 0.00 0.939 0.94  
## null.cach 5 792.33 5.48 0.061 1.00

^year is definitely best! null far above d2

#test for NB or Poisson - most should use Poisson ...   
testP.cach <- pcount(~1 ~1, cach.abund, mixture="P", K=12)  
testNB.cach <- pcount(~1 ~1, cach.abund, mixture="NB", K=12)  
fmsTEST <- fitList(testP.cach, testNB.cach)

## Warning in fitList(testP.cach, testNB.cach): Your list was unnamed, so  
## model names were added as object names

msTEST.cach <- modSel(fmsTEST)  
msTEST.cach

## nPars AIC delta AICwt cumltvWt  
## testP.cach 2 787.53 0.00 0.73 0.73  
## testNB.cach 3 789.53 2.00 0.27 1.00

## P is best for this species.

#date2 is better but time quadratic is not  
det.date.cach <- pcount(~ Jdate ~1, cach.abund, mixture="P", K=15)  
det.date2.cach <- pcount(~ Jdate + I(Jdate^2) ~1, cach.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.cach, det.date2.cach)

## Warning in fitList(det.date.cach, det.date2.cach): Your list was unnamed,  
## so model names were added as object names

DATEtest.cach <- modSel(mstestDATE)  
DATEtest.cach

## nPars AIC delta AICwt cumltvWt  
## det.date2.cach 4 788.52 0.00 0.62 0.62  
## det.date.cach 3 789.47 0.95 0.38 1.00

det.time.cach <-pcount(~ Time ~1, cach.abund, mixture="P",K=15)  
det.time2.cach <-pcount(~ Time + I(Time^2) ~1, cach.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.cach, det.time2.cach)

## Warning in fitList(det.time.cach, det.time2.cach): Your list was unnamed,  
## so model names were added as object names

TIMEtest.cach <- modSel(mstestTIME)  
TIMEtest.cach

## nPars AIC delta AICwt cumltvWt  
## det.time.cach 3 788.80 0.00 0.72 0.72  
## det.time2.cach 4 790.71 1.92 0.28 1.00

#detection covariates first  
det.null.cach <- pcount(~1 ~1, cach.abund, mixture="P", K=15)  
det.weather.cach <- pcount(~ Wind + Sky ~1, cach.abund, mixture="P", K=15)  
det.global2.cach <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise + Time ~1, cach.abund, mixture="P", K=15)  
det.sound.cach <- pcount(~ Noise + Wind ~1, cach.abund, mixture="P", K=15)  
det.date2.cach <- pcount(~ Jdate + I(Jdate^2) ~1, cach.abund, mixture="P", K=15)  
det.detect2.cach <- pcount(~ Jdate + I(Jdate^2) + Noise + Time ~1, cach.abund, mixture="P", K=15)  
det.notdate.cach <-pcount(~ Wind + Sky + Noise ~1, cach.abund, mixture="P", K=15)  
det.time.cach <-pcount(~ Time ~1, cach.abund, mixture="P",K=15)  
det.timing2.cach <-pcount(~ Time + Jdate + I(Jdate^2) ~1, cach.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.cach, det.weather.cach, det.global2.cach,  
 det.sound.cach, det.date2.cach, det.detect2.cach, det.notdate.cach,  
 det.time.cach, det.timing2.cach)

## Warning in fitList(det.null.cach, det.weather.cach, det.global2.cach,  
## det.sound.cach, : Your list was unnamed, so model names were added as  
## object names

msDC.cach <- modSel(fmsDC)  
msDC.cach

## nPars AIC delta AICwt cumltvWt  
## det.null.cach 2 787.52 0.00 0.277 0.28  
## det.date2.cach 4 788.52 1.00 0.168 0.45  
## det.time.cach 3 788.80 1.27 0.147 0.59  
## det.weather.cach 4 789.41 1.88 0.108 0.70  
## det.sound.cach 4 789.58 2.06 0.099 0.80  
## det.timing2.cach 5 789.76 2.24 0.091 0.89  
## det.notdate.cach 5 790.77 3.24 0.055 0.95  
## det.detect2.cach 6 791.62 4.10 0.036 0.98  
## det.global2.cach 8 792.88 5.35 0.019 1.00

#msDC.cach@Full  
#summary: null best, time second best, then weather, then date  
#summary NOW with quad: null, date2, time, weather

^ null best, date2, time, then weather

det.date2.cach #second best model - Jdate^2 is sig -

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ 1, data = cach.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.26 0.252 5.01 5.4e-07  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.76977 0.3743 -2.0565 0.0397  
## Jdate 0.00129 0.0689 0.0187 0.9851  
## I(Jdate^2) -0.11848 0.0707 -1.6760 0.0937  
##   
## AIC: 788.5212

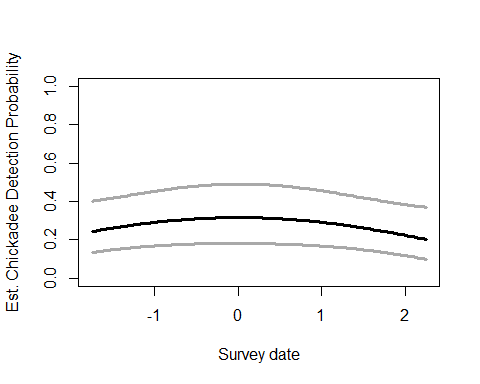
confint(det.date2.cach, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.5034155 -0.03612154  
## p(Jdate) -0.1337251 0.13630190  
## p(I(Jdate^2)) -0.2570292 0.02007641

confint(det.date2.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.767614 1.753683

det.date2.cach <- pcount(~ Jdate + I(Jdate^2) ~1, cach.abund, mixture="P", K=15)  
  
#summary(obsCovs(cach.abund))  
NewData.cach <-data.frame(Jdate=seq(-1.75,2.25,length=100))  
cach.est.date2 <- predict(det.date2.cach, type="det", newdata=NewData.cach,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=cach.est.date2, ylim=c(0,1), type="l", lwd=3,  
xlab="Survey date", ylab="Est. Chickadee Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=cach.est.date2, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=cach.est.date2, type="l", lwd=3, col="darkgray")



det.time.cach #third best model - time not sig

##   
## Call:  
## pcount(formula = ~Time ~ 1, data = cach.abund, K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.31 0.266 4.92 8.68e-07  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9498 0.3671 -2.588 0.00966  
## Time -0.0572 0.0674 -0.849 0.39607  
##   
## AIC: 788.7962

confint(det.time.cach, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.6692645 -0.23041008  
## p(Time) -0.1892945 0.07489847

confint(det.time.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.7863213 1.827832

det.weather.cach #4th best - non-sig variables

##   
## Call:  
## pcount(formula = ~Wind + Sky ~ 1, data = cach.abund, K = 15,   
## mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.34 0.277 4.84 1.29e-06  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9959 0.3781 -2.63 0.00845  
## Wind 0.0893 0.0708 1.26 0.20699  
## Sky -0.0597 0.0720 -0.83 0.40639  
##   
## AIC: 789.4061

confint(det.weather.cach, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.73707549 -0.25477124  
## p(Wind) -0.04941671 0.22807805  
## p(Sky) -0.20079207 0.08129623

confint(det.weather.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.7967447 1.880969

write.table(msDC.cach@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/cach\_top\_models\_msDC.xls",sep="\t")

# not BA, HWdens, FG\_herb, FG\_shrub, HW\_saplings  
testR.cach <- pcount(~1 ~BA, cach.abund, mixture="P", K=15)  
testQ.cach <- pcount(~1 ~BA + I(BA^2), cach.abund, mixture="P", K=15)  
msBAtest <- fitList(testR.cach, testQ.cach)

## Warning in fitList(testR.cach, testQ.cach): Your list was unnamed, so model  
## names were added as object names

BAtest.cach <- modSel(msBAtest)  
BAtest.cach

## nPars AIC delta AICwt cumltvWt  
## testR.cach 3 789.33 0.00 0.58 0.58  
## testQ.cach 4 789.97 0.64 0.42 1.00

testR.cach <- pcount(~1 ~HW\_dens\_1050, cach.abund, mixture="P", K=15)  
testQ.cach <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), cach.abund, mixture="P", K=15)  
msHW1050test <- fitList(testR.cach, testQ.cach)

## Warning in fitList(testR.cach, testQ.cach): Your list was unnamed, so model  
## names were added as object names

HW1050test.cach <- modSel(msHW1050test)  
HW1050test.cach

## nPars AIC delta AICwt cumltvWt  
## testR.cach 3 788.34 0.00 0.73 0.73  
## testQ.cach 4 790.33 1.99 0.27 1.00

testR.cach <- pcount(~1 ~FG\_herb, cach.abund, mixture="P", K=15)  
testQ.cach <- pcount(~1 ~FG\_herb + I(FG\_herb^2), cach.abund, mixture="P", K=15)  
msFGHtest <- fitList(testR.cach, testQ.cach)

## Warning in fitList(testR.cach, testQ.cach): Your list was unnamed, so model  
## names were added as object names

FGHtest.cach <- modSel(msFGHtest)  
FGHtest.cach

## nPars AIC delta AICwt cumltvWt  
## testR.cach 3 789.52 0.00 0.60 0.60  
## testQ.cach 4 790.34 0.82 0.40 1.00

testR.cach <- pcount(~1 ~FG\_shrub, cach.abund, mixture="P", K=15)  
testQ.cach <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), cach.abund, mixture="P", K=15)  
msFGStest <- fitList(testR.cach, testQ.cach)

## Warning in fitList(testR.cach, testQ.cach): Your list was unnamed, so model  
## names were added as object names

FGStest.cach <- modSel(msFGStest)  
FGStest.cach

## nPars AIC delta AICwt cumltvWt  
## testR.cach 3 789.52 0.00 0.69 0.69  
## testQ.cach 4 791.14 1.62 0.31 1.00

testR.cach <- pcount(~1 ~NHW\_saplings, cach.abund, mixture="P", K=15)  
testQ.cach <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), cach.abund, mixture="P", K=15)  
msHWStest <- fitList(testR.cach, testQ.cach)

## Warning in fitList(testR.cach, testQ.cach): Your list was unnamed, so model  
## names were added as object names

HWStest.cach <- modSel(msHWStest)  
HWStest.cach

## nPars AIC delta AICwt cumltvWt  
## testR.cach 3 787.30 0.00 0.71 0.71  
## testQ.cach 4 789.08 1.78 0.29 1.00

##site covariates next  
#null detection covariates (this was the best model)  
Nnull.cach <- pcount(~1 ~1  
 ,cach.abund, mixture="P", K=40)  
Nglobal.cach <- pcount(~ 1  
 ~ Treatment + Herbicide + BA + Nsnags +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + Rel\_HW2P\_canopy  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , cach.abund, mixture="P", K=40) #FPSiteIndex removed + HW\_dens\_1050 + FG\_herb + FG\_shrub + NHW\_saplings + NP\_over\_20cm   
Nlocal.cach <- pcount(~ 1  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , cach.abund, mixture="P", K=40)  
Nlh.cach <- pcount(~ 1  
 ~ TreeHt + Ccover + Nsnags + Rel\_HW2P\_canopy + YearCat  
 , cach.abund, mixture="P", K=40)  
Nlandmetrics.cach <- pcount (~ 1  
 ~ Parea + ShapeIndex + YearCat  
 , cach.abund, mixture="P",K=40)  
Nlandscape500.cach <- pcount(~ 1  
 ~ Evergreen500m + OpenDev500m + Schrubs500m + Ag500m + YearCat  
 , cach.abund, mixture="P", K=40)  
Nlandscape1.cach <- pcount(~ 1  
 ~ Evergreen1km + OpenDev1km + Schrubs1km + YearCat  
 , cach.abund, mixture="P", K=40)  
Nlandscape5.cach <- pcount(~ 1  
 ~ Evergreen5km + OpenDev5km + Schrubs1km + YearCat  
 , cach.abund, mixture="P", K=40)  
Nlandscape30.cach <- pcount(~ 1  
 ~ Evergreen30km + OpenDev30km + YearCat  
 , cach.abund, mixture="P", K=40)  
Ntreatment.cach <- pcount(~ 1 ~ Treatment + Nthins + YearCat  
 , cach.abund, mixture ="P", K=40)  
Nmanagement.cach <- pcount(~ 1 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , cach.abund, mixture="P", K=40)  
Ndisturbance.cach <- pcount(~ 1 ~ TimeSinceB + TimeSinceT + YearCat  
 , cach.abund, mixture="P", K=40)  
Nsiteprod.cach <- pcount(~ 1 ~ PISoils + NSoilTypes + YearCat  
 , cach.abund, mixture="P", K=40) #FPSiteIndex removed  
Nupstate.cach <- pcount(~ 1 ~ TreeHt + Ccover + Parea + YearCat, cach.abund, mixture="P", K=40)  
Ncoord.cach <- pcount (~1 ~ Latitude + Longitude + YearCat  
 , cach.abund, mixture="P", K=40)  
  
fmsN <- fitList(Nnull.cach, Nglobal.cach, Nlocal.cach, Nlh.cach,  
 Nlandmetrics.cach,  
 Nlandscape500.cach, Nlandscape1.cach, Nlandscape5.cach, Nlandscape30.cach,  
 Ntreatment.cach, Nmanagement.cach, Ndisturbance.cach,  
 Nsiteprod.cach, Nupstate.cach, Ncoord.cach)

## Warning in fitList(Nnull.cach, Nglobal.cach, Nlocal.cach, Nlh.cach,  
## Nlandmetrics.cach, : Your list was unnamed, so model names were added as  
## object names

msN.cach <- modSel(fmsN)  
#msN.cach@Full  
msN.cach

## nPars AIC delta AICwt cumltvWt  
## Nsiteprod.cach 5 783.45 0.00 2.0e-01 0.20  
## Nlandmetrics.cach 5 783.63 0.18 1.8e-01 0.38  
## Ndisturbance.cach 5 784.90 1.45 9.7e-02 0.48  
## Nupstate.cach 6 785.18 1.73 8.4e-02 0.56  
## Nlandscape1.cach 6 785.37 1.92 7.6e-02 0.64  
## Nlandscape500.cach 7 785.60 2.15 6.8e-02 0.71  
## Nlandscape30.cach 5 785.87 2.42 6.0e-02 0.77  
## Ncoord.cach 5 785.96 2.51 5.7e-02 0.82  
## Nlocal.cach 6 786.31 2.85 4.8e-02 0.87  
## Nlh.cach 7 786.99 3.54 3.4e-02 0.91  
## Ntreatment.cach 7 787.04 3.59 3.3e-02 0.94  
## Nlandscape5.cach 6 787.44 3.98 2.7e-02 0.97  
## Nnull.cach 2 787.52 4.07 2.6e-02 0.99  
## Nmanagement.cach 10 790.19 6.74 6.9e-03 1.00  
## Nglobal.cach 21 806.92 23.46 1.6e-06 1.00

#summary: siteprod best model, landmetrics next, then disturbance, upstate, & landscape1

# only significant variable is year in the next 5 models - no figures

Nsiteprod.cach

##   
## Call:  
## pcount(formula = ~1 ~ PISoils + NSoilTypes + YearCat, data = cach.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.6073 0.3472 4.629 3.67e-06  
## PISoils 0.0357 0.0660 0.541 5.88e-01  
## NSoilTypes -0.1015 0.0697 -1.456 1.45e-01  
## YearCatB -0.3856 0.1400 -2.755 5.87e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## -1.13 0.461 -2.46 0.0138  
##   
## AIC: 783.4526

confint(Nsiteprod.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.92683086 2.28781473  
## lam(PISoils) -0.09360798 0.16502335  
## lam(NSoilTypes) -0.23818005 0.03511073  
## lam(YearCatB) -0.65988232 -0.11126884

Nlandmetrics.cach

##   
## Call:  
## pcount(formula = ~1 ~ Parea + ShapeIndex + YearCat, data = cach.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.6132 0.3486 4.627 3.71e-06  
## Parea -0.0867 0.0767 -1.130 2.58e-01  
## ShapeIndex -0.0505 0.0705 -0.716 4.74e-01  
## YearCatB -0.3811 0.1399 -2.724 6.45e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## -1.15 0.463 -2.48 0.0133  
##   
## AIC: 783.629

confint(Nlandmetrics.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.9298729 2.29655145  
## lam(Parea) -0.2370203 0.06362522  
## lam(ShapeIndex) -0.1887761 0.08770690  
## lam(YearCatB) -0.6552254 -0.10688742

Ndisturbance.cach

##   
## Call:  
## pcount(formula = ~1 ~ TimeSinceB + TimeSinceT + YearCat, data = cach.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.5433 0.3082 5.008 5.49e-07  
## TimeSinceB -0.0483 0.0729 -0.663 5.07e-01  
## TimeSinceT -0.0524 0.0829 -0.632 5.27e-01  
## YearCatB -0.3877 0.1418 -2.735 6.24e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## -1.04 0.416 -2.51 0.0121  
##   
## AIC: 784.9019

confint(Ndisturbance.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.9393671 2.14730452  
## lam(TimeSinceB) -0.1911890 0.09455542  
## lam(TimeSinceT) -0.2149768 0.11012687  
## lam(YearCatB) -0.6655524 -0.10982780

Nupstate.cach

##   
## Call:  
## pcount(formula = ~1 ~ TreeHt + Ccover + Parea + YearCat, data = cach.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.6416 0.3543 4.633 3.61e-06  
## TreeHt -0.0460 0.0783 -0.587 5.57e-01  
## Ccover -0.0490 0.0675 -0.726 4.68e-01  
## Parea -0.0903 0.0759 -1.190 2.34e-01  
## YearCatB -0.4382 0.1556 -2.817 4.85e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## -1.15 0.471 -2.44 0.0148  
##   
## AIC: 785.1841

confint(Nupstate.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.9470823 2.33606235  
## lam(TreeHt) -0.1993560 0.10740931  
## lam(Ccover) -0.1812217 0.08329256  
## lam(Parea) -0.2390670 0.05845091  
## lam(YearCatB) -0.7430506 -0.13329440

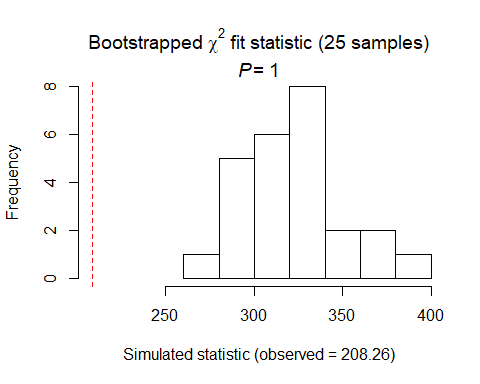
Nlandscape1.cach

##   
## Call:  
## pcount(formula = ~1 ~ Evergreen1km + OpenDev1km + Schrubs1km +   
## YearCat, data = cach.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.58837 0.3337 4.7602 1.93e-06  
## Evergreen1km 0.00105 0.0707 0.0149 9.88e-01  
## OpenDev1km 0.10140 0.0682 1.4861 1.37e-01  
## Schrubs1km 0.04277 0.0658 0.6497 5.16e-01  
## YearCatB -0.39092 0.1405 -2.7817 5.41e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## -1.11 0.445 -2.48 0.013  
##   
## AIC: 785.3727

confint(Nlandscape1.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.93437775 2.2423565  
## lam(Evergreen1km) -0.13760327 0.1397122  
## lam(OpenDev1km) -0.03233517 0.2351272  
## lam(Schrubs1km) -0.08625353 0.1717990  
## lam(YearCatB) -0.66635591 -0.1154809

write.table(msN.cach@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/cach\_top\_models\_msN.xls",sep="\t")



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 208.2609   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 279 311 320 337 397   
##   
## Estimate of c-hat = 0.64

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 208.2609

# using second best DC model (DATE^2 but not better than null)  
null.cach <- pcount(~ Jdate + I(Jdate^2) ~1, cach.abund, mixture="P", K=40)  
global.cach <- pcount(~ Jdate + I(Jdate^2)   
 ~ Treatment + Herbicide + BA + Nsnags +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + FG\_herb + FG\_shrub + NHW\_saplings  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , cach.abund, mixture="P", K=40)  
local.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , cach.abund, mixture="P", K=40) #can only include BA OR CCover  
lh.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ TreeHt + Ccover + Nsnags + Rel\_HW2P\_canopy + YearCat  
 , cach.abund, mixture="P", K=40)  
landmetrics.cach <- pcount (~ Jdate + I(Jdate^2)  
 ~ Parea + ShapeIndex + YearCat  
 , cach.abund, mixture="P",K=40)  
landscape500.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Evergreen500m + OpenDev500m + Schrubs500m + Ag500m + YearCat  
 , cach.abund, mixture="P", K=40)  
landscape1.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Evergreen1km + OpenDev1km + Schrubs1km + YearCat  
 , cach.abund, mixture="P", K=40)  
landscape5.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Evergreen5km + OpenDev5km + Schrubs1km + YearCat  
 , cach.abund, mixture="P", K=40)  
landscape30.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Evergreen30km + OpenDev30km + YearCat  
 , cach.abund, mixture="P", K=40)  
treatment.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + Nthins + YearCat  
 , cach.abund, mixture ="P", K=40)  
management.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , cach.abund, mixture="P", K=40)  
disturbance.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , cach.abund, mixture="P", K=40)  
siteprod.cach <- pcount(~ Jdate + I(Jdate^2) ~ PISoils + NSoilTypes + YearCat  
 , cach.abund, mixture="P", K=40) #FPSiteIndex removed  
upstate.cach <- pcount(~ Jdate + I(Jdate^2)  
 ~ TreeHt + Ccover + Parea + YearCat, cach.abund, mixture="P", K=40)  
coord.cach <- pcount (~Jdate + I(Jdate^2) ~ Latitude + Longitude + YearCat  
 , cach.abund, mixture="P", K=40)  
  
  
fmsCACH <- fitList(null.cach, global.cach, local.cach, lh.cach, landmetrics.cach,  
 landscape500.cach, landscape1.cach, landscape5.cach, landscape30.cach,  
 treatment.cach, management.cach, disturbance.cach,  
 siteprod.cach, upstate.cach, coord.cach)

## Warning in fitList(null.cach, global.cach, local.cach, lh.cach,  
## landmetrics.cach, : Your list was unnamed, so model names were added as  
## object names

ms.cach <- modSel(fmsCACH)  
ms.cach

## nPars AIC delta AICwt cumltvWt  
## landmetrics.cach 7 782.78 0.00 2.1e-01 0.21  
## siteprod.cach 7 783.03 0.25 1.8e-01 0.39  
## disturbance.cach 7 784.22 1.45 1.0e-01 0.49  
## upstate.cach 8 784.49 1.72 8.8e-02 0.58  
## landscape1.cach 8 784.68 1.91 8.0e-02 0.66  
## landscape500.cach 9 785.10 2.32 6.5e-02 0.72  
## landscape30.cach 7 785.21 2.44 6.1e-02 0.78  
## coord.cach 7 785.27 2.50 5.9e-02 0.84  
## local.cach 8 785.85 3.07 4.5e-02 0.89  
## treatment.cach 9 786.32 3.54 3.5e-02 0.92  
## lh.cach 9 786.53 3.76 3.2e-02 0.95  
## landscape5.cach 8 786.82 4.04 2.7e-02 0.98  
## null.cach 4 788.52 5.75 1.2e-02 0.99  
## management.cach 12 789.73 6.95 6.4e-03 1.00  
## global.cach 26 809.17 26.39 3.9e-07 1.00

#ms.cach@Full

^# diff results as with null - landmetrics best, siteprod next, disturbance 3rd, upstate 4th, landscape1 5th. But still, no variables significant except year, so no figures.

landmetrics.cach

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ Parea + ShapeIndex + YearCat,   
## data = cach.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.5982 0.3445 4.639 3.50e-06  
## Parea -0.0928 0.0772 -1.202 2.30e-01  
## ShapeIndex -0.0508 0.0710 -0.716 4.74e-01  
## YearCatB -0.4309 0.1420 -3.034 2.41e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.95986 0.4766 -2.0142 0.0440  
## Jdate 0.00399 0.0668 0.0597 0.9524  
## I(Jdate^2) -0.14468 0.0678 -2.1342 0.0328  
##   
## AIC: 782.7751

confint(landmetrics.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.9229475 2.27347305  
## lam(Parea) -0.2441961 0.05857894  
## lam(ShapeIndex) -0.1898579 0.08827017  
## lam(YearCatB) -0.7093106 -0.15258855

siteprod.cach

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ PISoils + NSoilTypes +   
## YearCat, data = cach.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.5714 0.3289 4.778 1.77e-06  
## PISoils 0.0324 0.0668 0.486 6.27e-01  
## NSoilTypes -0.0986 0.0703 -1.403 1.61e-01  
## YearCatB -0.4330 0.1425 -3.037 2.39e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.92432 0.4595 -2.0116 0.0443  
## Jdate 0.00633 0.0672 0.0943 0.9249  
## I(Jdate^2) -0.13977 0.0685 -2.0410 0.0412  
##   
## AIC: 783.0278

confint(Nsiteprod.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.92683086 2.28781473  
## lam(PISoils) -0.09360798 0.16502335  
## lam(NSoilTypes) -0.23818005 0.03511073  
## lam(YearCatB) -0.65988232 -0.11126884

disturbance.cach

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ TimeSinceB + TimeSinceT +   
## YearCat, data = cach.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.5242 0.2998 5.085 3.69e-07  
## TimeSinceB -0.0549 0.0735 -0.747 4.55e-01  
## TimeSinceT -0.0465 0.0829 -0.560 5.75e-01  
## YearCatB -0.4363 0.1440 -3.029 2.45e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.84988 0.4246 -2.001 0.0453  
## Jdate 0.00643 0.0677 0.095 0.9243  
## I(Jdate^2) -0.14470 0.0688 -2.103 0.0355  
##   
## AIC: 784.2245

confint(disturbance.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.9366650 2.11176879  
## lam(TimeSinceB) -0.1990312 0.08922465  
## lam(TimeSinceT) -0.2090260 0.11606948  
## lam(YearCatB) -0.7186410 -0.15397791

upstate.cach

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ TreeHt + Ccover + Parea +   
## YearCat, data = cach.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.6182 0.3455 4.684 2.81e-06  
## TreeHt -0.0411 0.0788 -0.521 6.02e-01  
## Ccover -0.0457 0.0677 -0.674 5.00e-01  
## Parea -0.0975 0.0765 -1.274 2.03e-01  
## YearCatB -0.4822 0.1572 -3.066 2.17e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.95519 0.4801 -1.9895 0.0466  
## Jdate 0.00338 0.0669 0.0505 0.9597  
## I(Jdate^2) -0.14280 0.0681 -2.0967 0.0360  
##   
## AIC: 784.4905

confint(upstate.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.9410848 2.29533348  
## lam(TreeHt) -0.1955881 0.11338874  
## lam(Ccover) -0.1783277 0.08702546  
## lam(Parea) -0.2473630 0.05244355  
## lam(YearCatB) -0.7903338 -0.17396650

landscape1.cach

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ Evergreen1km + OpenDev1km +   
## Schrubs1km + YearCat, data = cach.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.56369 0.3235 4.8336 1.34e-06  
## Evergreen1km 0.00357 0.0713 0.0501 9.60e-01  
## OpenDev1km 0.10407 0.0689 1.5096 1.31e-01  
## Schrubs1km 0.04267 0.0662 0.6444 5.19e-01  
## YearCatB -0.43925 0.1429 -3.0747 2.11e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9066 0.4528 -2.002 0.0452  
## Jdate 0.0085 0.0672 0.127 0.8993  
## I(Jdate^2) -0.1440 0.0684 -2.106 0.0352  
##   
## AIC: 784.6826

confint(landscape1.cach, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.92964149 2.1977480  
## lam(Evergreen1km) -0.13609510 0.1432319  
## lam(OpenDev1km) -0.03105073 0.2391904  
## lam(Schrubs1km) -0.08710855 0.1724408  
## lam(YearCatB) -0.71924817 -0.1592459

#write.table(ms.cch@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/cach\_top\_models\_ms.xls",sep="\t")

CACH summary: P distribution DCs: null best, date2 second best (Jdate^2 is - sig), time third best, then weather SCs, using null: Site prod best (- with YearB,) landmetrics next (- with YearB) then disturbance (- with YearB) then upstate (- with YearB) then landscape1 (- with YearB)

SCs, using Date^2: all same.

# EABL

eabl.abund<- csvToUMF("eabl\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(eabl.abund)  
#str(eabl.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(eabl.abund)= scale (obsCovs(eabl.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(eabl.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(eabl.abund) <- sc

#run this when have CSV with both years  
null.eabl <- pcount(~Jdate + Noise + Time ~1, eabl.abund, mixture="P", K=40)  
year.eabl <- pcount(~Jdate + Noise + Time ~ YearCat, eabl.abund, mixture="P", K=40)  
fms.year.eabl<- fitList(null.eabl, year.eabl)

## Warning in fitList(null.eabl, year.eabl): Your list was unnamed, so model  
## names were added as object names

year.ms.eabl<-modSel(fms.year.eabl)  
year.ms.eabl

## nPars AIC delta AICwt cumltvWt  
## null.eabl 5 513.47 0.00 0.69 0.69  
## year.eabl 6 515.05 1.58 0.31 1.00

^null best but year <d.2

#date is quad but not time  
det.date.eabl <- pcount(~ Jdate ~1, eabl.abund, mixture="P", K=15)  
det.date2.eabl <- pcount(~ Jdate + I(Jdate^2) ~1, eabl.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.eabl, det.date2.eabl)

## Warning in fitList(det.date.eabl, det.date2.eabl): Your list was unnamed,  
## so model names were added as object names

DATEtest.eabl <- modSel(mstestDATE)  
DATEtest.eabl

## nPars AIC delta AICwt cumltvWt  
## det.date2.eabl 4 509.53 0.00 0.54 0.54  
## det.date.eabl 3 509.83 0.29 0.46 1.00

det.time.eabl <-pcount(~ Time ~1, eabl.abund, mixture="P",K=15)  
det.time2.eabl <-pcount(~ Time + I(Time^2) ~1, eabl.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.eabl, det.time2.eabl)

## Warning in fitList(det.time.eabl, det.time2.eabl): Your list was unnamed,  
## so model names were added as object names

TIMEtest.eabl <- modSel(mstestTIME)  
TIMEtest.eabl

## nPars AIC delta AICwt cumltvWt  
## det.time.eabl 3 511.91 0.00 0.69 0.69  
## det.time2.eabl 4 513.53 1.62 0.31 1.00

#detection covariates first  
det.null.eabl <- pcount(~1 ~1, eabl.abund, mixture="P", K=15)  
det.weather.eabl <- pcount(~ Wind + Sky ~1, eabl.abund, mixture="P", K=15)  
det.global2.eabl <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time ~1, eabl.abund, mixture="P", K=15)  
det.sound.eabl <- pcount(~ Noise + Wind ~1, eabl.abund, mixture="P", K=15)  
det.date2.eabl <- pcount(~ Jdate + I(Jdate^2) ~1, eabl.abund, mixture="P", K=15)  
det.detect2.eabl <- pcount(~ Jdate + I(Jdate^2) + Noise + Time ~1, eabl.abund, mixture="P", K=15)  
det.notdate.eabl <-pcount(~ Wind + Sky + Noise ~1, eabl.abund, mixture="P", K=15)  
det.time.eabl <-pcount(~ Time ~1, eabl.abund, mixture="P",K=15)  
det.timing2.eabl <-pcount(~ Time + Jdate + I(Jdate^2) ~1, eabl.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.eabl, det.weather.eabl, det.global2.eabl,  
 det.sound.eabl, det.date2.eabl, det.detect2.eabl, det.notdate.eabl,  
 det.time.eabl, det.timing2.eabl)

## Warning in fitList(det.null.eabl, det.weather.eabl, det.global2.eabl,  
## det.sound.eabl, : Your list was unnamed, so model names were added as  
## object names

msDC.eabl <- modSel(fmsDC)  
msDC.eabl

## nPars AIC delta AICwt cumltvWt  
## det.date2.eabl 4 509.53 0.00 0.3315 0.33  
## det.null.eabl 2 509.92 0.39 0.2734 0.60  
## det.timing2.eabl 5 511.26 1.72 0.1401 0.74  
## det.time.eabl 3 511.91 2.38 0.1007 0.85  
## det.detect2.eabl 6 513.17 3.63 0.0539 0.90  
## det.sound.eabl 4 513.80 4.27 0.0392 0.94  
## det.weather.eabl 4 513.82 4.29 0.0388 0.98  
## det.notdate.eabl 5 515.76 6.23 0.0147 0.99  
## det.global2.eabl 8 517.03 7.50 0.0078 1.00

#msDC.eabl@Full  
#summary: date is best, closely followed by null, timing is also <2 (didn't change)

det.date2.eabl #non-sig with Jdate

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ 1, data = eabl.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.194 0.175 1.11 0.267  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.414 0.293 -1.41 0.1578  
## Jdate 0.194 0.114 1.70 0.0892  
## I(Jdate^2) -0.170 0.114 -1.50 0.1339  
##   
## AIC: 509.5325

confint(det.date2.eabl, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.98729677 0.16029111  
## p(Jdate) -0.02975594 0.41853649  
## p(I(Jdate^2)) -0.39344106 0.05245958

confint(det.date2.eabl, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.1483556 0.5365231

det.timing2.eabl #non-sig with either

##   
## Call:  
## pcount(formula = ~Time + Jdate + I(Jdate^2) ~ 1, data = eabl.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.189 0.173 1.09 0.275  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.3991 0.292 -1.366 0.1719  
## Time 0.0617 0.118 0.525 0.5995  
## Jdate 0.2103 0.119 1.772 0.0764  
## I(Jdate^2) -0.1746 0.114 -1.527 0.1267  
##   
## AIC: 511.2554

confint(det.timing2.eabl, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.97155941 0.17345557  
## p(Time) -0.16874380 0.29223816  
## p(Jdate) -0.02233637 0.44299882  
## p(I(Jdate^2)) -0.39874637 0.04946605

confint(det.timing2.eabl, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.1504421 0.5283466

write.table(msDC.eabl@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/eabl\_top\_models\_msDC.xls",sep="\t")

#BA IS quad! also HW\_dens, FG\_herb  
#not FG\_shrub, NHW\_saplings  
testR.eabl <- pcount(~1 ~BA, eabl.abund, mixture="P", K=4)  
testQ.eabl <- pcount(~1 ~BA + I(BA^2), eabl.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.eabl, testQ.eabl)

## Warning in fitList(testR.eabl, testQ.eabl): Your list was unnamed, so model  
## names were added as object names

BAtest.eabl <- modSel(msBAtest)  
BAtest.eabl

## nPars AIC delta AICwt cumltvWt  
## testQ.eabl 4 503.57 0.00 0.65 0.65  
## testR.eabl 3 504.77 1.21 0.35 1.00

testR.eabl <- pcount(~1 ~HW\_dens\_1050, eabl.abund, mixture="P", K=4)  
testQ.eabl <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), eabl.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.eabl, testQ.eabl)

## Warning in fitList(testR.eabl, testQ.eabl): Your list was unnamed, so model  
## names were added as object names

HW1050test.eabl <- modSel(msHW1050test)  
HW1050test.eabl

## nPars AIC delta AICwt cumltvWt  
## testQ.eabl 4 508.20 0.00 0.72 0.72  
## testR.eabl 3 510.09 1.89 0.28 1.00

testR.eabl <- pcount(~1 ~FG\_herb, eabl.abund, mixture="P", K=4)  
testQ.eabl <- pcount(~1 ~FG\_herb + I(FG\_herb^2), eabl.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.eabl, testQ.eabl)

## Warning in fitList(testR.eabl, testQ.eabl): Your list was unnamed, so model  
## names were added as object names

FGHtest.eabl <- modSel(msFGHtest)  
FGHtest.eabl

## nPars AIC delta AICwt cumltvWt  
## testQ.eabl 4 511.78 0.00 0.53 0.53  
## testR.eabl 3 511.98 0.20 0.47 1.00

#not  
testR.eabl <- pcount(~1 ~FG\_shrub, eabl.abund, mixture="P", K=4)  
testQ.eabl <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), eabl.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.eabl, testQ.eabl)

## Warning in fitList(testR.eabl, testQ.eabl): Your list was unnamed, so model  
## names were added as object names

FGStest.eabl <- modSel(msFGStest)  
FGStest.eabl

## nPars AIC delta AICwt cumltvWt  
## testR.eabl 3 511.29 0.00 0.72 0.72  
## testQ.eabl 4 513.21 1.92 0.28 1.00

#not  
testR.eabl <- pcount(~1 ~NHW\_saplings, eabl.abund, mixture="P", K=4)  
testQ.eabl <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), eabl.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.eabl, testQ.eabl)

## Warning in fitList(testR.eabl, testQ.eabl): Your list was unnamed, so model  
## names were added as object names

HWStest.eabl <- modSel(msHWStest)  
HWStest.eabl

## nPars AIC delta AICwt cumltvWt  
## testR.eabl 3 512.53 0.00 0.63 0.63  
## testQ.eabl 4 513.58 1.05 0.37 1.00

#site covariates using detection covariates (date)  
null.eabl <- pcount(~ Jdate + I(Jdate^2) ~1, eabl.abund, mixture="P", K=80)  
global.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + Herbicide + BA + I(BA^2) + Nsnags +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + I(HW\_dens\_1050^2) + FG\_herb + I(FG\_herb^2)  
 + NHW\_saplings  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , eabl.abund, mixture="P", K=80)  
local.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , eabl.abund, mixture="P", K=80) #can only include BA OR CCover  
lh.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ BA + I(BA^2) + FG\_herb + I(FG\_herb^2)   
 + HW\_dens\_1050 + I(HW\_dens\_1050^2) + NHW\_saplings + Nsnags + YearCat  
 , eabl.abund, mixture="P", K=80)  
# covariates: BA, canopy cover, low height and mid-height veg (lack thereof), cavities  
landmetrics.eabl <- pcount (~ Jdate + I(Jdate^2)  
 ~ Parea + ShapeIndex + YearCat  
 , eabl.abund, mixture="P",K=80)  
landscape500.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass500m + OpenDev500m + Schrubs500m + Ag500m + YearCat  
 , eabl.abund, mixture="P", K=80)  
landscape1.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass1km + OpenDev1km + Schrubs1km + Ag1km + YearCat  
 , eabl.abund, mixture="P", K=80)  
landscape5.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass5km + OpenDev5km + Schrubs5km + YearCat  
 , eabl.abund, mixture="P", K=80)  
landscape30.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass30km + OpenDev30km + YearCat  
 , eabl.abund, mixture="P", K=80)  
treatment.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + Nthins + YearCat  
 , eabl.abund, mixture ="P", K=80)  
management.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + BA + I(BA^2) + TimeSinceB + TimeSinceT  
 + Herbicide + YearCat  
 , eabl.abund, mixture="P", K=80)  
disturbance.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , eabl.abund, mixture="P", K=80)  
siteprod.eabl <- pcount(~ Jdate + I(Jdate^2) ~ PISoils + NSoilTypes + YearCat  
 , eabl.abund, mixture="P", K=80) # + FPSiteIndex  
#upstate.eabl <- pcount(~ Jdate + I(Jdate^2) ~ X + Y + Z, eabl.abund, mixture="P", K=80)  
coord.eabl <- pcount (~Jdate + I(Jdate^2) ~ Latitude + Longitude + YearCat  
 , eabl.abund, mixture="P", K=80)  
  
fmsEABL <- fitList(null.eabl, global.eabl, local.eabl, lh.eabl, landmetrics.eabl,  
 landscape500.eabl, landscape1.eabl, landscape5.eabl, landscape30.eabl,  
 treatment.eabl, management.eabl, disturbance.eabl,  
 siteprod.eabl, coord.eabl)

## Warning in fitList(null.eabl, global.eabl, local.eabl, lh.eabl,  
## landmetrics.eabl, : Your list was unnamed, so model names were added as  
## object names

ms.eabl <- modSel(fmsEABL) #note this does not include upstate  
ms.eabl

## nPars AIC delta AICwt cumltvWt  
## lh.eabl 13 502.95 0.00 7.4e-01 0.74  
## landmetrics.eabl 7 505.68 2.73 1.9e-01 0.92  
## null.eabl 4 509.53 6.58 2.7e-02 0.95  
## local.eabl 8 510.29 7.34 1.9e-02 0.97  
## management.eabl 13 511.42 8.47 1.1e-02 0.98  
## landscape1.eabl 9 512.79 9.84 5.4e-03 0.99  
## coord.eabl 7 513.23 10.28 4.3e-03 0.99  
## landscape30.eabl 7 514.13 11.18 2.7e-03 0.99  
## landscape5.eabl 8 514.28 11.33 2.6e-03 0.99  
## disturbance.eabl 7 514.62 11.67 2.1e-03 1.00  
## siteprod.eabl 7 514.74 11.79 2.0e-03 1.00  
## landscape500.eabl 9 517.01 14.06 6.5e-04 1.00  
## treatment.eabl 9 518.91 15.96 2.5e-04 1.00  
## global.eabl 28 524.57 21.62 1.5e-05 1.00

#ms.eabl@Full  
#LH best and only under 2.0 (land metrics pushed out) (didnt change with quads)

lh.eabl

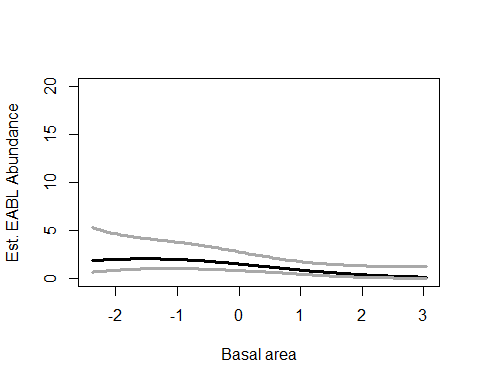
##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ BA + I(BA^2) + FG\_herb +   
## I(FG\_herb^2) + HW\_dens\_1050 + I(HW\_dens\_1050^2) + NHW\_saplings +   
## Nsnags + YearCat, data = eabl.abund, K = 80, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.4046 0.3125 1.295 0.1954  
## BA -0.4078 0.1575 -2.590 0.0096  
## I(BA^2) -0.1350 0.0968 -1.394 0.1634  
## FG\_herb -0.0567 0.1436 -0.395 0.6929  
## I(FG\_herb^2) -0.1596 0.1234 -1.293 0.1959  
## HW\_dens\_1050 0.0438 0.1730 0.253 0.8003  
## I(HW\_dens\_1050^2) 0.1374 0.0880 1.561 0.1185  
## NHW\_saplings -0.1422 0.1207 -1.178 0.2388  
## Nsnags 0.2082 0.0977 2.131 0.0331  
## YearCatB 0.0285 0.2342 0.122 0.9031  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.774 0.382 -2.02 0.043  
## Jdate 0.163 0.110 1.48 0.138  
## I(Jdate^2) -0.149 0.109 -1.37 0.171  
##   
## AIC: 502.9501

confint(lh.eabl, type="state",method="normal")

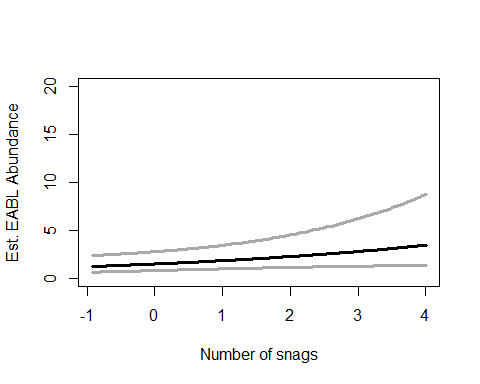
## 0.025 0.975  
## lam(Int) -0.20789209 1.01712936  
## lam(BA) -0.71636862 -0.09917512  
## lam(I(BA^2)) -0.32477061 0.05482305  
## lam(FG\_herb) -0.33814745 0.22471983  
## lam(I(FG\_herb^2)) -0.40155500 0.08228460  
## lam(HW\_dens\_1050) -0.29534098 0.38287591  
## lam(I(HW\_dens\_1050^2)) -0.03507451 0.30978944  
## lam(NHW\_saplings) -0.37876827 0.09436985  
## lam(Nsnags) 0.01671552 0.39970761  
## lam(YearCatB) -0.43051173 0.48755944

LH (- with BA, + with Nsnags)

#life history (-BA)  
lh.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ BA + I(BA^2) + FG\_herb + I(FG\_herb^2)   
 + HW\_dens\_1050 + I(HW\_dens\_1050^2) + NHW\_saplings + Nsnags + YearCat  
 , eabl.abund, mixture="P", K=80)  
  
ND.eabl <-data.frame(BA=seq(min(sc$BA),max(sc$BA),length=100),FG\_herb=0, HW\_dens\_1050=0,NHW\_saplings=0,Nsnags=0,YearCat=0)  
eabl.est.ba <- predict(lh.eabl, type="state",  
newdata=ND.eabl,appendData=TRUE)  
  
plot(Predicted~ BA, data=eabl.est.ba, ylim=c(0,20), type="l", lwd=3,  
xlab="Basal area", ylab="Est. EABL Abundance")  
##95% confidence intervals  
lines(lower~ BA, data=eabl.est.ba, type="l", lwd=3, col="darkgray")  
lines(upper~ BA, data=eabl.est.ba, type="l", lwd=3, col="darkgray")

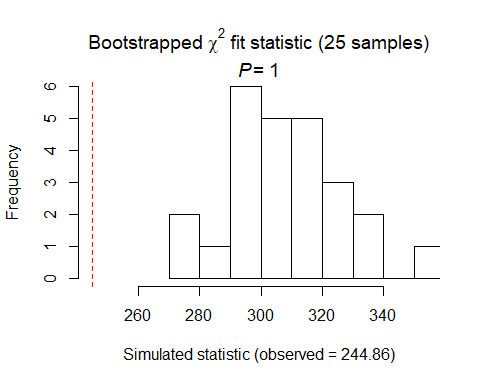


#life history (+Nsnags)  
lh.eabl <- pcount(~ Jdate + I(Jdate^2)  
 ~ BA + I(BA^2) + FG\_herb + I(FG\_herb^2)   
 + HW\_dens\_1050 + I(HW\_dens\_1050^2) + NHW\_saplings + Nsnags + YearCat  
 , eabl.abund, mixture="P", K=80)  
  
ND2.eabl <-data.frame(Nsnags=seq(min(sc$Nsnags),max(sc$Nsnags),length=100),FG\_herb=0, HW\_dens\_1050=0,NHW\_saplings=0,BA=0,YearCat=0)  
eabl.est.snags <- predict(lh.eabl, type="state",  
newdata=ND2.eabl,appendData=TRUE)  
  
plot(Predicted~ Nsnags, data=eabl.est.snags, ylim=c(0,20), type="l", lwd=3,  
xlab="Number of snags", ylab="Est. EABL Abundance")  
##95% confidence intervals  
lines(lower~ Nsnags, data=eabl.est.snags, type="l", lwd=3, col="darkgray")  
lines(upper~ Nsnags, data=eabl.est.snags, type="l", lwd=3, col="darkgray")



write.table(ms.eabl@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/XXX\_top\_models\_msX.xls",sep="\t")

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 244.856   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 274 295 307 314 354   
##   
## Estimate of c-hat = 0.8

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 244.856

EABL summary P distribution SCs: date2, null, timing all under 2.0 (non sig with any) DCs using date2 model: LH only top model (- with BA, + with Nsnags)

# EAWP

# EAWP (forests (prefer deciduious or more edgy?), tree-nester, flycatching, insectivore)

# covariates: less canopy cover? less dense stand? hardwoody?

eawp.abund<- csvToUMF("eawp\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(eawp.abund)  
#str(eawp.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(eawp.abund)= scale (obsCovs(eawp.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(eawp.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(eawp.abund) <- sc

#run this when have CSV with both years  
null.eawp <- pcount(~Jdate + Noise + Time ~1, eawp.abund, mixture="P", K=40)  
year.eawp <- pcount(~Jdate + Noise + Time ~ YearCat, eawp.abund, mixture="P", K=40)  
fms.year.eawp<- fitList(null.eawp, year.eawp)

## Warning in fitList(null.eawp, year.eawp): Your list was unnamed, so model  
## names were added as object names

year.ms.eawp<-modSel(fms.year.eawp)  
year.ms.eawp

## nPars AIC delta AICwt cumltvWt  
## null.eawp 5 596.87 0.00 0.68 0.68  
## year.eawp 6 598.34 1.47 0.32 1.00

^ null ranked higher but year 1.47 <2

#neither of these better quadratic  
det.date.eawp <- pcount(~ Jdate ~1, eawp.abund, mixture="P", K=15)  
det.date2.eawp <- pcount(~ Jdate + I(Jdate^2) ~1, eawp.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.eawp, det.date2.eawp)

## Warning in fitList(det.date.eawp, det.date2.eawp): Your list was unnamed,  
## so model names were added as object names

DATEtest.eawp <- modSel(mstestDATE)  
DATEtest.eawp

## nPars AIC delta AICwt cumltvWt  
## det.date.eawp 3 594.28 0.00 0.63 0.63  
## det.date2.eawp 4 595.38 1.10 0.37 1.00

det.time.eawp <-pcount(~ Time ~1, eawp.abund, mixture="P",K=15)  
det.time2.eawp <-pcount(~ Time + I(Time^2) ~1, eawp.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.eawp, det.time2.eawp)

## Warning in fitList(det.time.eawp, det.time2.eawp): Your list was unnamed,  
## so model names were added as object names

TIMEtest.eawp <- modSel(mstestTIME)  
TIMEtest.eawp

## nPars AIC delta AICwt cumltvWt  
## det.time.eawp 3 593.84 0.00 0.62 0.62  
## det.time2.eawp 4 594.82 0.98 0.38 1.00

#detection covariates first  
det.null.eawp <- pcount(~1 ~1, eawp.abund, mixture="P", K=15)  
det.weather.eawp <- pcount(~ Wind + Sky ~1, eawp.abund, mixture="P", K=15)  
det.global.eawp <- pcount(~ Jdate + Wind + Sky + Noise +Time ~1, eawp.abund, mixture="P", K=15)  
det.sound.eawp <- pcount(~ Noise + Wind ~1, eawp.abund, mixture="P", K=15)  
det.date.eawp <- pcount(~ Jdate ~1, eawp.abund, mixture="P", K=15)  
det.detect.eawp <- pcount(~ Jdate + Noise + Time ~1, eawp.abund, mixture="P", K=15)  
det.notdate.eawp <-pcount(~ Wind + Sky + Noise ~1, eawp.abund, mixture="P", K=15)  
det.time.eawp <-pcount(~ Time ~1, eawp.abund, mixture="P",K=15)  
det.timing.eawp <-pcount(~ Time + Jdate ~1, eawp.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.eawp, det.weather.eawp, det.global.eawp,  
 det.sound.eawp, det.date.eawp, det.detect.eawp, det.notdate.eawp,  
 det.time.eawp, det.timing.eawp)

## Warning in fitList(det.null.eawp, det.weather.eawp, det.global.eawp,  
## det.sound.eawp, : Your list was unnamed, so model names were added as  
## object names

msDC.eawp <- modSel(fmsDC)  
msDC.eawp

## nPars AIC delta AICwt cumltvWt  
## det.weather.eawp 4 592.28 0.00 0.242 0.24  
## det.null.eawp 2 592.46 0.18 0.221 0.46  
## det.time.eawp 3 593.84 1.56 0.111 0.57  
## det.notdate.eawp 5 594.01 1.73 0.102 0.68  
## det.date.eawp 3 594.28 2.00 0.089 0.76  
## det.global.eawp 7 594.42 2.15 0.083 0.85  
## det.sound.eawp 4 594.56 2.28 0.077 0.93  
## det.timing.eawp 4 595.41 3.13 0.051 0.98  
## det.detect.eawp 5 596.87 4.59 0.024 1.00

#msDC.ybch@Full  
#summary: weather best model (wind, sky) then null then time then notdate <2

det.weather.eawp

##   
## Call:  
## pcount(formula = ~Wind + Sky ~ 1, data = eawp.abund, K = 15,   
## mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.275 0.123 2.23 0.0255  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.0222 0.199 -0.112 0.911  
## Wind -0.1493 0.111 -1.341 0.180  
## Sky 0.1771 0.109 1.619 0.105  
##   
## AIC: 592.2759

confint(det.weather.eawp, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.41248214 0.36806052  
## p(Wind) -0.36752496 0.06886577  
## p(Sky) -0.03723459 0.39145868

confint(det.weather.eawp, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.03375248 0.5164475

write.table(msDC.eawp@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/eawp\_top\_models\_msDC.xls",sep="\t")

#none of these better quadratic  
testR.eawp <- pcount(~1 ~BA, eawp.abund, mixture="P", K=4)  
testQ.eawp <- pcount(~1 ~BA + I(BA^2), eawp.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.eawp, testQ.eawp)

## Warning in fitList(testR.eawp, testQ.eawp): Your list was unnamed, so model  
## names were added as object names

BAtest.eawp <- modSel(msBAtest)  
BAtest.eawp

## nPars AIC delta AICwt cumltvWt  
## testR.eawp 3 588.26 0.00 0.73 0.73  
## testQ.eawp 4 590.25 1.98 0.27 1.00

testR.eawp <- pcount(~1 ~HW\_dens\_1050, eawp.abund, mixture="P", K=4)  
testQ.eawp <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), eawp.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.eawp, testQ.eawp)

## Warning in fitList(testR.eawp, testQ.eawp): Your list was unnamed, so model  
## names were added as object names

HW1050test.eawp <- modSel(msHW1050test)  
HW1050test.eawp

## nPars AIC delta AICwt cumltvWt  
## testR.eawp 3 590.80 0.00 0.65 0.65  
## testQ.eawp 4 592.01 1.21 0.35 1.00

testR.eawp <- pcount(~1 ~FG\_herb, eawp.abund, mixture="P", K=4)  
testQ.eawp <- pcount(~1 ~FG\_herb + I(FG\_herb^2), eawp.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.eawp, testQ.eawp)

## Warning in fitList(testR.eawp, testQ.eawp): Your list was unnamed, so model  
## names were added as object names

FGHtest.eawp <- modSel(msFGHtest)  
FGHtest.eawp

## nPars AIC delta AICwt cumltvWt  
## testR.eawp 3 592.76 0.00 0.71 0.71  
## testQ.eawp 4 594.54 1.78 0.29 1.00

testR.eawp <- pcount(~1 ~FG\_shrub, eawp.abund, mixture="P", K=4)  
testQ.eawp <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), eawp.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.eawp, testQ.eawp)

## Warning in fitList(testR.eawp, testQ.eawp): Your list was unnamed, so model  
## names were added as object names

FGStest.eawp <- modSel(msFGStest)  
FGStest.eawp

## nPars AIC delta AICwt cumltvWt  
## testR.eawp 3 594.64 0.00 0.58 0.58  
## testQ.eawp 4 595.32 0.68 0.42 1.00

testR.eawp <- pcount(~1 ~NHW\_saplings, eawp.abund, mixture="P", K=4)  
testQ.eawp <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), eawp.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.eawp, testQ.eawp)

## Warning in fitList(testR.eawp, testQ.eawp): Your list was unnamed, so model  
## names were added as object names

HWStest.eawp <- modSel(msHWStest)  
HWStest.eawp

## nPars AIC delta AICwt cumltvWt  
## testR.eawp 3 594.48 0.00 0.66 0.66  
## testQ.eawp 4 595.84 1.36 0.34 1.00

#more appropriate detection covariates (weather - wind + sky)  
null.eawp <- pcount(~ Wind + Sky ~1, eawp.abund, mixture="P", K=40)  
global.eawp <- pcount(~ Wind + Sky  
 ~ Treatment + Herbicide + BA +Ccover  
 + Ldepth + TreeHt + TimeSinceB + TimeSinceT + Nthins  
 + NHW\_saplings + Rel\_HW2P\_canopy  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , eawp.abund, mixture="P", K=40)  
#took out: Nsnags, Age, FPSiteIndex + HW\_dens\_1050 + FG\_herb + FG\_shrub + NP\_over\_20cm  
local.eawp <- pcount(~ Wind + Sky  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , eawp.abund, mixture="P", K=40) #can only include BA OR CCover  
lh.eawp <- pcount(~ Wind + Sky  
 ~ Rel\_HW2P\_canopy + Ccover + NHW\_saplings + YearCat  
 , eawp.abund, mixture="P", K=40)  
landmetrics.eawp <- pcount (~ Wind + Sky  
 ~ Parea + ShapeIndex + YearCat  
 , eawp.abund, mixture="P",K=40)  
landscape500.eawp <- pcount(~ Wind + Sky  
 ~ Evergreen500m + Ag500m + HighDev500m + YearCat  
 , eawp.abund, mixture="P", K=40)  
landscape1.eawp <- pcount(~ Wind + Sky  
 ~ Evergreen1km + HighDev1km + YearCat  
 , eawp.abund, mixture="P", K=40)  
landscape5.eawp <- pcount(~ Wind + Sky  
 ~ Evergreen5km + HighDev5km + YearCat  
 , eawp.abund, mixture="P", K=40)  
landscape30.eawp <- pcount(~ Wind + Sky  
 ~ Evergreen30km + HighDev30km + YearCat  
 , eawp.abund, mixture="P", K=40) #protected removed  
treatment.eawp <- pcount(~ Wind + Sky  
 ~ Treatment + Nthins + YearCat  
 , eawp.abund, mixture ="P", K=40)  
management.eawp <- pcount(~ Wind + Sky  
 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , eawp.abund, mixture="P", K=40)  
disturbance.eawp <- pcount(~ Wind + Sky  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , eawp.abund, mixture="P", K=40)  
siteprod.eawp <- pcount(~ Wind + Sky ~ PISoils + NSoilTypes + YearCat  
 , eawp.abund, mixture="P", K=40) #FPSiteIndex out  
#upstate.eawp <- pcount(~ Wind + Sky ~ X + Y + Z, eawp.abund, mixture="P", K=40)  
coord.eawp <- pcount (~ Wind + Sky ~ Latitude + Longitude + YearCat  
 , eawp.abund, mixture="P", K=40)  
  
fms <- fitList(null.eawp, global.eawp, local.eawp, lh.eawp, landmetrics.eawp,  
 landscape500.eawp, landscape1.eawp, landscape5.eawp, landscape30.eawp,  
 treatment.eawp, management.eawp, disturbance.eawp,  
 siteprod.eawp, coord.eawp)

## Warning in fitList(null.eawp, global.eawp, local.eawp, lh.eawp,  
## landmetrics.eawp, : Your list was unnamed, so model names were added as  
## object names

ms.eawp <- modSel(fms) #note this does not include upstate or FPSiteIndex  
ms.eawp

## nPars AIC delta AICwt cumltvWt  
## treatment.eawp 9 586.81 0.00 0.42450 0.42  
## disturbance.eawp 7 588.48 1.67 0.18448 0.61  
## landscape500.eawp 8 588.87 2.06 0.15180 0.76  
## management.eawp 12 589.58 2.77 0.10641 0.87  
## landscape5.eawp 7 591.90 5.09 0.03333 0.90  
## local.eawp 8 592.23 5.42 0.02827 0.93  
## null.eawp 4 592.28 5.47 0.02759 0.96  
## lh.eawp 8 593.04 6.23 0.01884 0.98  
## landscape1.eawp 7 593.98 7.17 0.01178 0.99  
## coord.eawp 7 595.70 8.89 0.00498 0.99  
## landscape30.eawp 7 596.65 9.84 0.00310 1.00  
## siteprod.eawp 7 596.94 10.13 0.00268 1.00  
## landmetrics.eawp 7 597.49 10.69 0.00203 1.00  
## global.eawp 22 601.88 15.07 0.00023 1.00

#ms.eawp@Full  
#summary: treatment, disturbance

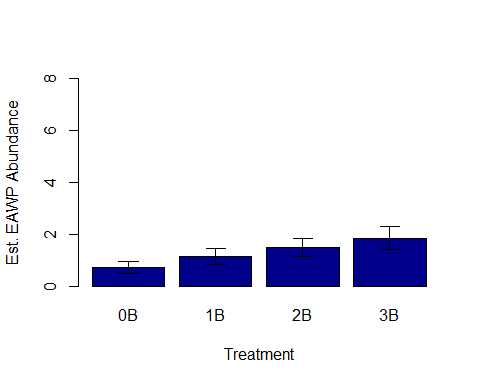
treatment.eawp

##   
## Call:  
## pcount(formula = ~Wind + Sky ~ Treatment + Nthins + YearCat,   
## data = eawp.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.33761 0.319 -1.0588 0.2897  
## Treatment1B 0.47063 0.372 1.2636 0.2064  
## Treatment2B 0.74097 0.351 2.1091 0.0349  
## Treatment3B 0.95130 0.370 2.5725 0.0101  
## Nthins 0.08504 0.107 0.7936 0.4274  
## YearCatB 0.00397 0.198 0.0201 0.9840  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.127 0.227 -0.559 0.576  
## Wind -0.147 0.111 -1.321 0.186  
## Sky 0.169 0.107 1.582 0.114  
##   
## AIC: 586.8088

confint(treatment.eawp, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.9626065 0.2873787  
## lam(Treatment1B) -0.2593672 1.2006292  
## lam(Treatment2B) 0.0523871 1.4295497  
## lam(Treatment3B) 0.2265158 1.6760856  
## lam(Nthins) -0.1249862 0.2950739  
## lam(YearCatB) -0.3833404 0.3912847

#treatment model (treatment variable)  
treatment.eawp <- pcount(~ Wind + Sky  
 ~ Treatment + Nthins + YearCat  
 , eawp.abund, mixture ="P", K=40)  
  
##  
error.bar <- function(x, y, upper, lower=upper, length=0.1,...){  
 if(length(x) != length(y) | length(y) !=length(lower) | length(lower) != length(upper))  
 stop("vectors must be same length")  
 arrows(x,y+upper, x, y-lower, angle=90, code=3, length=length, ...)  
}  
##  
  
ND.eawp <- data.frame(Treatment=factor(c("0B","1B","2B","3B")),  
 Nthins=0, YearCat=0)  
eawp.est.tx <- predict(treatment.eawp, type="state",  
 newdata=ND.eawp,appendData=TRUE)  
  
data.mean<-eawp.est.tx$Predicted  
data.sd<-eawp.est.tx$SE  
  
bar.p <-barplot(data.mean,  
 names.arg=c("0B","1B","2B","3B"),  
 ylim = c(0, 8), ylab="Est. EAWP Abundance", xlab="Treatment",  
 #cex.names = 1.5, cex.axis=1.5, cex.lab=1.5,   
 col="darkblue")  
error.bar(bar.p,data.mean,data.sd) #sd



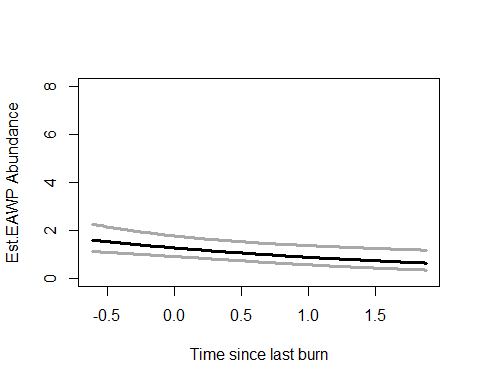
disturbance.eawp

##   
## Call:  
## pcount(formula = ~Wind + Sky ~ TimeSinceB + TimeSinceT + YearCat,   
## data = eawp.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.2379 0.169 1.406 0.15978  
## TimeSinceB -0.3667 0.131 -2.790 0.00527  
## TimeSinceT 0.0946 0.104 0.914 0.36084  
## YearCatB 0.0400 0.196 0.203 0.83878  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.0953 0.220 -0.434 0.664  
## Wind -0.1477 0.112 -1.324 0.185  
## Sky 0.1609 0.107 1.501 0.133  
##   
## AIC: 588.4756

confint(disturbance.eawp, type="state",method="normal")

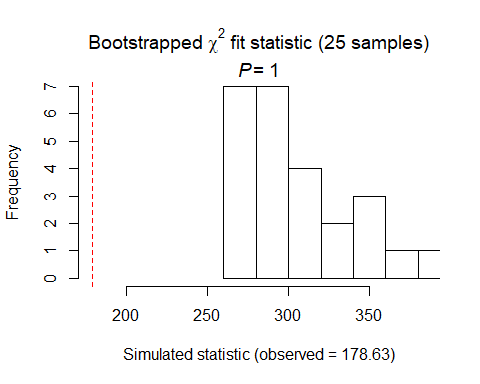
## 0.025 0.975  
## lam(Int) -0.09375929 0.5694852  
## lam(TimeSinceB) -0.62426313 -0.1090747  
## lam(TimeSinceT) -0.10830305 0.2974900  
## lam(YearCatB) -0.34514116 0.4250940

#disturbance model with timesinceburn  
disturbance.eawp <- pcount(~ Wind + Sky  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , eawp.abund, mixture="P", K=40)  
  
ND2.eawp <-data.frame(TimeSinceB=seq(min(sc$TimeSinceB),max(sc$TimeSinceB),length=100),TimeSinceT=0,YearCat=0)  
eawp.est.timeb <- predict(disturbance.eawp, type="state",  
newdata=ND2.eawp,appendData=TRUE)  
  
plot(Predicted~ TimeSinceB, data=eawp.est.timeb, ylim=c(0,8), type="l", lwd=3,  
xlab="Time since last burn", ylab="Est.EAWP Abundance")  
##95% confidence intervals  
lines(lower~ TimeSinceB, data=eawp.est.timeb, type="l", lwd=3, col="darkgray")  
lines(upper~ TimeSinceB, data=eawp.est.timeb, type="l", lwd=3, col="darkgray")



write.table(ms.eawp@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/eawp\_top\_models\_ms.xls",sep="\t")

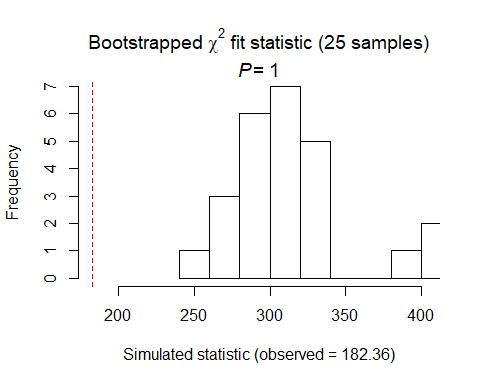
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 178.6298   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 265 278 296 324 385   
##   
## Estimate of c-hat = 0.59

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 178.6298

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 182.3636   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 258 291 315 325 403   
##   
## Estimate of c-hat = 0.58

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 182.3636

EAWP Summary P distribution nothing quadratic DCs: weather (none actually sig), null, time, notdate SCs using weather: Treatment (+ with 2B, + with 3B) Disturbance (- with TimeSinceB)

# INBU

# (foliage gleaner, shrub-nesting, insects, open woodland habitat & EDGES)

# covariates: grasses, shrub density, within a meter above ground, low branches, tree age

inbu.abund<- csvToUMF("inbu\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(inbu.abund)  
#str(inbu.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(inbu.abund)= scale (obsCovs(inbu.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(inbu.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)]) #from 26 to 74 +landscape+soils  
siteCovs(inbu.abund) <- sc

#run this when have CSV with both years  
null.inbu <- pcount(~Jdate + Noise + Time ~1, inbu.abund, mixture="P", K=40)  
year.inbu <- pcount(~Jdate + Noise + Time ~ YearCat, inbu.abund, mixture="P", K=40)  
fms.year.inbu<- fitList(null.inbu, year.inbu)

## Warning in fitList(null.inbu, year.inbu): Your list was unnamed, so model  
## names were added as object names

year.ms.inbu<-modSel(fms.year.inbu)  
year.ms.inbu

## nPars AIC delta AICwt cumltvWt  
## year.inbu 6 621.62 0.00 0.968 0.97  
## null.inbu 5 628.43 6.81 0.032 1.00

^ definite year effect - null model is way above 2.0

det.date.inbu <- pcount(~ Jdate ~1, inbu.abund, mixture="P", K=15)  
det.date2.inbu <- pcount(~ Jdate + I(Jdate^2) ~1, inbu.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.inbu, det.date2.inbu)

## Warning in fitList(det.date.inbu, det.date2.inbu): Your list was unnamed,  
## so model names were added as object names

DATEtest.inbu <- modSel(mstestDATE)  
DATEtest.inbu

## nPars AIC delta AICwt cumltvWt  
## det.date.inbu 3 630.60 0.00 0.66 0.66  
## det.date2.inbu 4 631.89 1.29 0.34 1.00

det.time.inbu <-pcount(~ Time ~1, inbu.abund, mixture="P",K=15)  
det.time2.inbu <-pcount(~ Time + I(Time^2) ~1, inbu.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.inbu, det.time2.inbu)

## Warning in fitList(det.time.inbu, det.time2.inbu): Your list was unnamed,  
## so model names were added as object names

TIMEtest.inbu <- modSel(mstestTIME)  
TIMEtest.inbu

## nPars AIC delta AICwt cumltvWt  
## det.time2.inbu 4 645.74 0.00 0.59 0.59  
## det.time.inbu 3 646.49 0.75 0.41 1.00

#detection covariates first  
det.null.inbu <- pcount(~1 ~1, inbu.abund, mixture="P", K=15)  
det.weather.inbu <- pcount(~ Wind + Sky ~1, inbu.abund, mixture="P", K=15)  
det.global2.inbu <- pcount(~ Jdate + Wind + Sky + Noise +Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
det.sound.inbu <- pcount(~ Noise + Wind ~1, inbu.abund, mixture="P", K=15)  
det.date.inbu <- pcount(~ Jdate ~1, inbu.abund, mixture="P", K=15)  
det.detect2.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
det.notdate.inbu <-pcount(~ Wind + Sky + Noise ~1, inbu.abund, mixture="P", K=15)  
det.time2.inbu <-pcount(~ Time + I(Time^2) ~1, inbu.abund, mixture="P",K=15)  
det.timing2.inbu <-pcount(~ Time + I(Time^2) + Jdate ~1, inbu.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.inbu, det.weather.inbu, det.global2.inbu,  
 det.sound.inbu, det.date.inbu, det.detect2.inbu, det.notdate.inbu,  
 det.time2.inbu, det.timing2.inbu)

## Warning in fitList(det.null.inbu, det.weather.inbu, det.global2.inbu,  
## det.sound.inbu, : Your list was unnamed, so model names were added as  
## object names

msDC.inbu <- modSel(fmsDC)  
msDC.inbu

## nPars AIC delta AICwt cumltvWt  
## det.detect2.inbu 6 626.45 0.00 5.1e-01 0.51  
## det.global2.inbu 8 627.47 1.02 3.1e-01 0.82  
## det.timing2.inbu 5 629.36 2.91 1.2e-01 0.94  
## det.date.inbu 3 630.60 4.15 6.4e-02 1.00  
## det.time2.inbu 4 645.74 19.29 3.3e-05 1.00  
## det.sound.inbu 4 645.98 19.53 2.9e-05 1.00  
## det.null.inbu 2 646.59 20.14 2.2e-05 1.00  
## det.notdate.inbu 5 647.78 21.33 1.2e-05 1.00  
## det.weather.inbu 4 649.98 23.54 4.0e-06 1.00

#msDC.inbu@Full  
#summary: detect (Date+Noise+Time) first, then global (same)

det.detect2.inbu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time + I(Time^2) ~ 1, data = inbu.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.425 0.124 3.43 0.000609  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.476 0.213 -2.235 2.54e-02  
## Jdate 0.455 0.108 4.228 2.36e-05  
## Noise -0.248 0.112 -2.207 2.73e-02  
## Time -0.108 0.118 -0.919 3.58e-01  
## I(Time^2) 0.224 0.114 1.967 4.91e-02  
##   
## AIC: 626.4477

confint(det.detect2.inbu, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.8939064378 -0.05862764  
## p(Jdate) 0.2441711042 0.66623734  
## p(Noise) -0.4684869591 -0.02780233  
## p(Time) -0.3399173910 0.12297867  
## p(I(Time^2)) 0.0008475647 0.44668437

confint(det.detect2.inbu, type="state",method="normal")

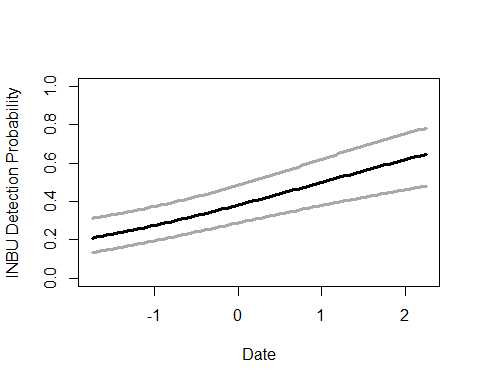
## 0.025 0.975  
## lam(Int) 0.1820665 0.6684174

^ Detect best model (+Date, +Time2, -Noise), global next best model

#detect2 model - jdate variable - averaged with global model too  
det.global2.inbu <- pcount(~ Jdate + Wind + Sky + Noise +Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
det.detect2.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
  
#summary(inbu.abund)  
dms\_top.inbu <- fitList(det.detect2.inbu,det.global2.inbu)

## Warning in fitList(det.detect2.inbu, det.global2.inbu): Your list was  
## unnamed, so model names were added as object names

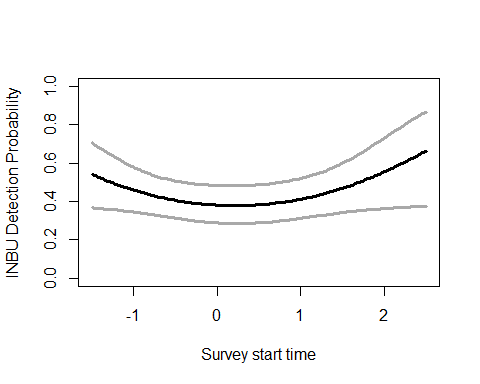
ND.inbud <-data.frame(Jdate=seq(-1.75,2.25,length=100),Noise=0,Time=0, Wind=0, Sky=0)  
inbu.est.date <- predict(dms\_top.inbu, type="det",  
 newdata=ND.inbud,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=inbu.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Date", ylab="INBU Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=inbu.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=inbu.est.date, type="l", lwd=3, col="darkgray")



#detect2 model - TIME variable - averaged with global model too  
det.global2.inbu <- pcount(~ Jdate + Wind + Sky + Noise +Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
det.detect2.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
  
#summary(obsCovs(inbu.abund))  
dms\_top.inbu <- fitList(det.detect2.inbu,det.global2.inbu)

## Warning in fitList(det.detect2.inbu, det.global2.inbu): Your list was  
## unnamed, so model names were added as object names

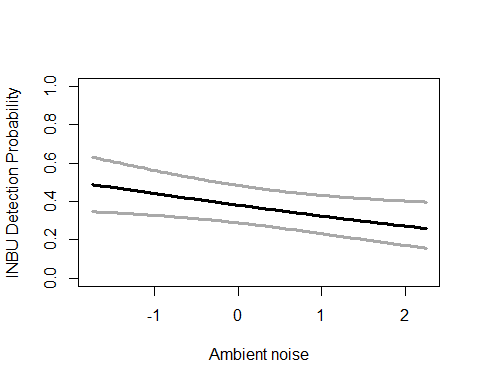
ND.inbud2 <-data.frame(Time=seq(-1.5,2.5,length=100),Noise=0,Jdate=0, Wind=0, Sky=0)  
inbu.est.time <- predict(dms\_top.inbu, type="det",  
 newdata=ND.inbud2,appendData=TRUE)  
  
plot(Predicted~ Time, data=inbu.est.time, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey start time", ylab="INBU Detection Probability")  
##95% confidence intervals  
lines(lower~ Time, data=inbu.est.time, type="l", lwd=3, col="darkgray")  
lines(upper~ Time, data=inbu.est.time, type="l", lwd=3, col="darkgray")



#detect2 model - NOISE variable - averaged with global model too  
det.global2.inbu <- pcount(~ Jdate + Wind + Sky + Noise +Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
det.detect2.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2) ~1, inbu.abund, mixture="P", K=15)  
  
#summary(obsCovs(inbu.abund))  
dms\_top.inbu <- fitList(det.detect2.inbu,det.global2.inbu)

## Warning in fitList(det.detect2.inbu, det.global2.inbu): Your list was  
## unnamed, so model names were added as object names

ND.inbud3 <-data.frame(Noise=seq(-1.75,2.25,length=100),Time=0,Jdate=0, Wind=0, Sky=0)  
inbu.est.noise <- predict(dms\_top.inbu, type="det",  
 newdata=ND.inbud3,appendData=TRUE)  
  
plot(Predicted~ Noise, data=inbu.est.noise, ylim=c(0,1), type="l", lwd=3,  
 xlab="Ambient noise", ylab="INBU Detection Probability")  
##95% confidence intervals  
lines(lower~ Noise, data=inbu.est.noise, type="l", lwd=3, col="darkgray")  
lines(upper~ Noise, data=inbu.est.noise, type="l", lwd=3, col="darkgray")



write.table(msDC.inbu@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/inbu\_top\_models\_msDC.xls",sep="\t")

#not using for now  
#predict(det.detect.inbu, type="det") #gave me 204 rows...  
#backTransform(det.detect.inbu, "psi", method=normal)

#none of these better quadratic  
testR.inbu <- pcount(~1 ~BA, inbu.abund, mixture="P", K=4)  
testQ.inbu <- pcount(~1 ~BA + I(BA^2), inbu.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.inbu, testQ.inbu)

## Warning in fitList(testR.inbu, testQ.inbu): Your list was unnamed, so model  
## names were added as object names

BAtest.inbu <- modSel(msBAtest)  
BAtest.inbu

## nPars AIC delta AICwt cumltvWt  
## testR.inbu 3 649.68 0.00 0.68 0.68  
## testQ.inbu 4 651.14 1.46 0.32 1.00

testR.inbu <- pcount(~1 ~HW\_dens\_1050, inbu.abund, mixture="P", K=4)  
testQ.inbu <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), inbu.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.inbu, testQ.inbu)

## Warning in fitList(testR.inbu, testQ.inbu): Your list was unnamed, so model  
## names were added as object names

HW1050test.inbu <- modSel(msHW1050test)  
HW1050test.inbu

## nPars AIC delta AICwt cumltvWt  
## testR.inbu 3 648.65 0.00 0.73 0.73  
## testQ.inbu 4 650.65 2.00 0.27 1.00

testR.inbu <- pcount(~1 ~FG\_herb, inbu.abund, mixture="P", K=4)  
testQ.inbu <- pcount(~1 ~FG\_herb + I(FG\_herb^2), inbu.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.inbu, testQ.inbu)

## Warning in fitList(testR.inbu, testQ.inbu): Your list was unnamed, so model  
## names were added as object names

FGHtest.inbu <- modSel(msFGHtest)  
FGHtest.inbu

## nPars AIC delta AICwt cumltvWt  
## testR.inbu 3 648.45 0.00 0.55 0.55  
## testQ.inbu 4 648.84 0.39 0.45 1.00

testR.inbu <- pcount(~1 ~FG\_shrub, inbu.abund, mixture="P", K=4)  
testQ.inbu <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), inbu.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.inbu, testQ.inbu)

## Warning in fitList(testR.inbu, testQ.inbu): Your list was unnamed, so model  
## names were added as object names

FGStest.inbu <- modSel(msFGStest)  
FGStest.inbu

## nPars AIC delta AICwt cumltvWt  
## testR.inbu 3 651.05 0.00 0.72 0.72  
## testQ.inbu 4 652.95 1.90 0.28 1.00

testR.inbu <- pcount(~1 ~NHW\_saplings, inbu.abund, mixture="P", K=4)  
testQ.inbu <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), inbu.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.inbu, testQ.inbu)

## Warning in fitList(testR.inbu, testQ.inbu): Your list was unnamed, so model  
## names were added as object names

HWStest.inbu <- modSel(msHWStest)  
HWStest.inbu

## nPars AIC delta AICwt cumltvWt  
## testR.inbu 3 649.78 0.00 0.70 0.70  
## testQ.inbu 4 651.53 1.74 0.30 1.00

#more appropriate detection covariates (detect best model)  
null.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2) ~1, inbu.abund, mixture="P", K=40)  
global.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Treatment + Herbicide + BA +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + FG\_herb + FG\_shrub + NHW\_saplings  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , inbu.abund, mixture="P", K=40)  
#doesn't include FPSiteIndex + NP\_over\_20cm + Rel\_HW2P\_canopy + Nsnags  
local.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , inbu.abund, mixture="P", K=40) #can only include BA OR CCover  
lh.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ HW\_dens\_1050 + FG\_herb + FG\_shrub + Age + NHW\_saplings + YearCat  
 , inbu.abund, mixture="P", K=40)  
landmetrics.inbu <- pcount (~ Jdate + Noise + Time + I(Time^2)  
 ~ Parea + ShapeIndex + YearCat  
 , inbu.abund, mixture="P",K=40)  
landscape500.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Evergreen500m + Grass500m + Ag500m + HighDev500m  
 + Schrubs500m + YearCat  
 , inbu.abund, mixture="P", K=40)  
landscape1.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Grass1km + Ag1km + HighDev1km + Schrubs1km + YearCat  
 , inbu.abund, mixture="P", K=40)  
landscape5.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Evergreen5km + Grass5km + HighDev5km + Schrubs5km + YearCat  
 , inbu.abund, mixture="P", K=40)  
landscape30.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Evergreen30km + Grass30km + HighDev30km + YearCat  
 , inbu.abund, mixture="P", K=40) #removed Protected  
treatment.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Treatment + Nthins + YearCat  
 , inbu.abund, mixture ="P", K=40)  
management.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , inbu.abund, mixture="P", K=40)  
disturbance.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , inbu.abund, mixture="P", K=40)  
siteprod.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ PISoils + NSoilTypes + YearCat  
 , inbu.abund, mixture="P", K=40)  
#upstate n/a no data  
coord.inbu <- pcount (~Jdate + Noise + Time + I(Time^2)  
 ~ Latitude + Longitude + YearCat  
 , inbu.abund, mixture="P", K=80)  
  
  
fms <- fitList(null.inbu, global.inbu, local.inbu, lh.inbu, landmetrics.inbu,  
 landscape500.inbu, landscape1.inbu, landscape5.inbu, landscape30.inbu,  
 treatment.inbu, management.inbu, disturbance.inbu,  
 siteprod.inbu, coord.inbu)

## Warning in fitList(null.inbu, global.inbu, local.inbu, lh.inbu,  
## landmetrics.inbu, : Your list was unnamed, so model names were added as  
## object names

ms.inbu <- modSel(fms) #note this does not include upstate or some of site.prod  
ms.inbu

## nPars AIC delta AICwt cumltvWt  
## local.inbu 10 617.91 0.00 2.8e-01 0.28  
## lh.inbu 12 618.82 0.91 1.8e-01 0.45  
## disturbance.inbu 9 618.83 0.92 1.7e-01 0.63  
## treatment.inbu 11 618.90 0.99 1.7e-01 0.79  
## siteprod.inbu 9 620.34 2.43 8.2e-02 0.87  
## coord.inbu 9 622.31 4.40 3.1e-02 0.91  
## landscape500.inbu 12 622.48 4.57 2.8e-02 0.93  
## landmetrics.inbu 9 623.26 5.36 1.9e-02 0.95  
## management.inbu 14 623.56 5.65 1.6e-02 0.97  
## landscape30.inbu 10 624.05 6.15 1.3e-02 0.98  
## landscape1.inbu 11 624.38 6.47 1.1e-02 0.99  
## null.inbu 6 626.45 8.54 3.9e-03 1.00  
## landscape5.inbu 11 626.59 8.69 3.6e-03 1.00  
## global.inbu 27 636.59 18.69 2.4e-05 1.00

#ms.inbu@Full  
#summary: local best, treatment second, disturbance third, life history 4th, siteprod also d<2

^local best (), treatment second, disturbance third, life history 4th, siteprod also d<2 ^^with updated quad time: local, life history, disturbance, treatment…

write.table(ms.inbu@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/inbu\_top\_models\_ms.xls",sep="\t")

local.inbu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time + I(Time^2) ~ Ccover +   
## TreeHt + Ldepth + YearCat, data = inbu.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.7934 0.1658 4.786 1.70e-06  
## Ccover -0.0471 0.0915 -0.514 6.07e-01  
## TreeHt -0.0529 0.1043 -0.507 6.12e-01  
## Ldepth -0.2486 0.1100 -2.260 2.38e-02  
## YearCatB -0.7619 0.2180 -3.494 4.76e-04  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.606 0.235 -2.584 9.78e-03  
## Jdate 0.434 0.106 4.092 4.28e-05  
## Noise -0.191 0.113 -1.699 8.93e-02  
## Time -0.100 0.115 -0.875 3.81e-01  
## I(Time^2) 0.234 0.112 2.097 3.60e-02  
##   
## AIC: 617.9083

confint(local.inbu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.4684679 1.11827800  
## lam(Ccover) -0.2264210 0.13231243  
## lam(TreeHt) -0.2572418 0.15143142  
## lam(Ldepth) -0.4641136 -0.03304592  
## lam(YearCatB) -1.1892219 -0.33448382

lh.inbu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time + I(Time^2) ~ HW\_dens\_1050 +   
## FG\_herb + FG\_shrub + Age + NHW\_saplings + YearCat, data = inbu.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.775586 0.1690 4.58921 4.45e-06  
## HW\_dens\_1050 0.124597 0.1077 1.15690 2.47e-01  
## FG\_herb 0.156687 0.1118 1.40204 1.61e-01  
## FG\_shrub 0.125676 0.0977 1.28572 1.99e-01  
## Age -0.000481 0.0918 -0.00524 9.96e-01  
## NHW\_saplings -0.149686 0.1098 -1.36292 1.73e-01  
## YearCatB -0.718517 0.2127 -3.37742 7.32e-04  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.637 0.244 -2.609 9.07e-03  
## Jdate 0.423 0.106 3.995 6.47e-05  
## Noise -0.174 0.113 -1.543 1.23e-01  
## Time -0.108 0.113 -0.949 3.42e-01  
## I(Time^2) 0.230 0.110 2.078 3.77e-02  
##   
## AIC: 618.8164

confint(lh.inbu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.44434843 1.10682424  
## lam(HW\_dens\_1050) -0.08648947 0.33568301  
## lam(FG\_herb) -0.06235185 0.37572505  
## lam(FG\_shrub) -0.06590591 0.31725801  
## lam(Age) -0.18043865 0.17947611  
## lam(NHW\_saplings) -0.36494450 0.06557202  
## lam(YearCatB) -1.13548242 -0.30155091

disturbance.inbu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time + I(Time^2) ~ TimeSinceB +   
## TimeSinceT + YearCat, data = inbu.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.688 0.154 4.474 7.69e-06  
## TimeSinceB -0.206 0.111 -1.863 6.25e-02  
## TimeSinceT -0.055 0.118 -0.466 6.41e-01  
## YearCatB -0.582 0.192 -3.038 2.38e-03  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.538 0.223 -2.407 1.61e-02  
## Jdate 0.437 0.107 4.066 4.78e-05  
## Noise -0.225 0.112 -2.011 4.43e-02  
## Time -0.100 0.116 -0.859 3.90e-01  
## I(Time^2) 0.216 0.113 1.911 5.60e-02  
##   
## AIC: 618.8289

confint(disturbance.inbu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.3863932 0.98896789  
## lam(TimeSinceB) -0.4231875 0.01073046  
## lam(TimeSinceT) -0.2860410 0.17606195  
## lam(YearCatB) -0.9571494 -0.20647574

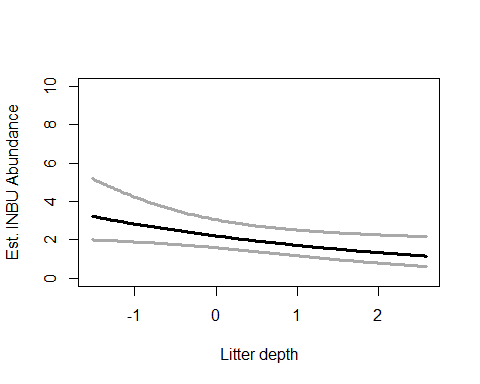
treatment.inbu

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time + I(Time^2) ~ Treatment +   
## Nthins + YearCat, data = inbu.abund, K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 0.2412 0.279 0.864 0.38764  
## Treatment1B 0.4724 0.308 1.532 0.12561  
## Treatment2B 0.3159 0.322 0.982 0.32619  
## Treatment3B 0.8575 0.334 2.564 0.01035  
## Nthins -0.0862 0.117 -0.736 0.46153  
## YearCatB -0.6136 0.194 -3.170 0.00153  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.5464 0.226 -2.413 0.015816  
## Jdate 0.4384 0.107 4.085 0.000044  
## Noise -0.2102 0.113 -1.864 0.062277  
## Time -0.0979 0.116 -0.843 0.399034  
## I(Time^2) 0.2112 0.113 1.861 0.062716  
##   
## AIC: 618.9033

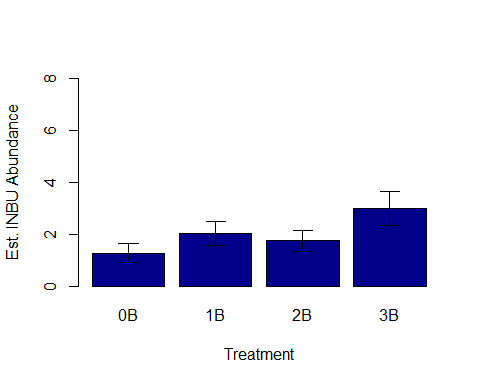
confint(treatment.inbu, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.3060400 0.7884792  
## lam(Treatment1B) -0.1321080 1.0769641  
## lam(Treatment2B) -0.3147167 0.9465072  
## lam(Treatment3B) 0.2020152 1.5129242  
## lam(Nthins) -0.3157495 0.1432931  
## lam(YearCatB) -0.9931088 -0.2341802

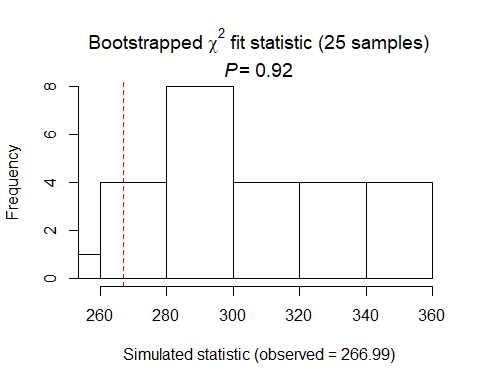
#site productivity with # soil types  
local.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , inbu.abund, mixture="P", K=40)  
  
ND.inbu2 <-data.frame(Ldepth=seq(min(sc$Ldepth),max(sc$Ldepth),length=100),Ccover=0,TreeHt=0,YearCat=0)  
inbu.est.litter <- predict(local.inbu, type="state",  
newdata=ND.inbu2,appendData=TRUE)  
  
plot(Predicted~ Ldepth, data=inbu.est.litter, ylim=c(0,10), type="l", lwd=3,  
xlab="Litter depth", ylab="Est. INBU Abundance")  
##95% confidence intervals  
lines(lower~ Ldepth, data=inbu.est.litter, type="l", lwd=3, col="darkgray")  
lines(upper~ Ldepth, data=inbu.est.litter, type="l", lwd=3, col="darkgray")



#treatment model (treatment variable)  
treatment.inbu <- pcount(~ Jdate + Noise + Time + I(Time^2)  
 ~ Treatment + Nthins + YearCat  
 , inbu.abund, mixture ="P", K=40)  
  
##  
error.bar <- function(x, y, upper, lower=upper, length=0.1,...){  
 if(length(x) != length(y) | length(y) !=length(lower) | length(lower) != length(upper))  
 stop("vectors must be same length")  
 arrows(x,y+upper, x, y-lower, angle=90, code=3, length=length, ...)  
}  
##  
  
ND.inbu <- data.frame(Treatment=factor(c("0B","1B","2B","3B")),  
 Nthins=0, YearCat=0)  
inbu.est.tx <- predict(treatment.inbu, type="state",  
 newdata=ND.inbu,appendData=TRUE)  
  
data.mean<-inbu.est.tx$Predicted  
data.sd<-inbu.est.tx$SE  
  
bar.p <-barplot(data.mean,  
 names.arg=c("0B","1B","2B","3B"),  
 ylim = c(0, 8), ylab="Est. INBU Abundance", xlab="Treatment",  
 #cex.names = 1.5, cex.axis=1.5, cex.lab=1.5,   
 col="darkblue")  
error.bar(bar.p,data.mean,data.sd) #sd



## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 266.995   
## Number of bootstrap samples = 25  
## P-value = 0.92  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 258 283 293 336 358   
##   
## Estimate of c-hat = 0.88

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 266.995

### not using yet  
#backTransform(linearComb(disturbance.inbu, coefficients = c(1,0), type= "det"))  
##YAY!  
#backTransform(linearComb(disturbance.inbu, coefficients = c(1,0,0), type= "state"))  
  
#or:  
#newData.inbu<-data.frame(TimeSinceT=0, TimeSinceB=0:20)  
#round(predict(disturbance.inbu, type="state", newdata=newData.inbu, appendData=TRUE, 2))  
##or "det"  
##confint(disturbance.inbu,type="det")

INBU Summary: P distribution DCs: detect2 best (- noise, + date, +-+ time2), global second SCs using detect2 model: with updated quad time: local, life history, disturbance, treatment… local best (- with Ldepth, - with YearB) life history 4th (- with YearB), disturbance third (- with YearB), treatment second (+ with Treatment 3B, - with YearB)

# NOBO

# (grasslands - open woodlands, ground nester, ground forager, plants)

# covariates: low BA, fires/disturbance, litter, low-medium HW cover?

nobo.abund<- csvToUMF("nobo\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(nobo.abund)  
#str(nobo.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(nobo.abund)= scale (obsCovs(nobo.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(nobo.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(nobo.abund) <- sc

#test for NB or Poisson - most should use Poisson ...   
testP.nobo <- pcount(~1 ~1, nobo.abund, mixture="P", K=4)  
testNB.nobo <- pcount(~1 ~1, nobo.abund, mixture="NB", K=4)  
fmsTEST <- fitList(testP.nobo, testNB.nobo)

## Warning in fitList(testP.nobo, testNB.nobo): Your list was unnamed, so  
## model names were added as object names

msTEST.nobo <- modSel(fmsTEST)  
msTEST.nobo

## nPars AIC delta AICwt cumltvWt  
## testNB.nobo 3 479.98 0.00 0.927 0.93  
## testP.nobo 2 485.06 5.08 0.073 1.00

## NB is best for this species. Change below to correspond!

#run this when have CSV with both years  
null.nobo <- pcount(~Jdate + Noise + Time ~1, nobo.abund, mixture="NB", K=40)  
year.nobo <- pcount(~Jdate + Noise + Time ~ YearCat, nobo.abund, mixture="NB", K=40)  
fms.year.nobo<- fitList(null.nobo, year.nobo)

## Warning in fitList(null.nobo, year.nobo): Your list was unnamed, so model  
## names were added as object names

year.ms.nobo<-modSel(fms.year.nobo)  
year.ms.nobo

## nPars AIC delta AICwt cumltvWt  
## year.nobo 7 433.60 0.00 0.85 0.85  
## null.nobo 6 437.08 3.48 0.15 1.00

^ year definite effect, null is much over d2

#date definite effect, not time  
det.date.nobo <- pcount(~ Jdate ~1, nobo.abund, mixture="P", K=15)  
det.date2.nobo <- pcount(~ Jdate + I(Jdate^2) ~1, nobo.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.nobo, det.date2.nobo)

## Warning in fitList(det.date.nobo, det.date2.nobo): Your list was unnamed,  
## so model names were added as object names

DATEtest.nobo <- modSel(mstestDATE)  
DATEtest.nobo

## nPars AIC delta AICwt cumltvWt  
## det.date2.nobo 4 437.47 0.00 0.988 0.99  
## det.date.nobo 3 446.24 8.76 0.012 1.00

det.time.nobo <-pcount(~ Time ~1, nobo.abund, mixture="P",K=15)  
det.time2.nobo <-pcount(~ Time + I(Time^2) ~1, nobo.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.nobo, det.time2.nobo)

## Warning in fitList(det.time.nobo, det.time2.nobo): Your list was unnamed,  
## so model names were added as object names

TIMEtest.nobo <- modSel(mstestTIME)  
TIMEtest.nobo

## nPars AIC delta AICwt cumltvWt  
## det.time.nobo 3 484.53 0.00 0.73 0.73  
## det.time2.nobo 4 486.49 1.96 0.27 1.00

#detection covariates first  
det.null.nobo <- pcount(~1 ~1, nobo.abund, mixture="NB", K=15)  
det.weather.nobo <- pcount(~ Wind + Sky ~1, nobo.abund, mixture="NB", K=15)  
det.global2.nobo <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time ~1, nobo.abund, mixture="NB", K=15)  
det.sound.nobo <- pcount(~ Noise + Wind ~1, nobo.abund, mixture="NB", K=15)  
det.date2.nobo <- pcount(~ Jdate + I(Jdate^2) ~1, nobo.abund, mixture="NB", K=15)  
det.detect2.nobo <- pcount(~ Jdate + I(Jdate^2) + Noise + Time ~1, nobo.abund, mixture="NB", K=15)  
det.notdate.nobo <-pcount(~ Wind + Sky + Noise ~1, nobo.abund, mixture="NB", K=15)  
det.time.nobo <-pcount(~ Time ~1, nobo.abund, mixture="NB",K=15)  
det.timing2.nobo <-pcount(~ Time + Jdate + I(Jdate^2) ~1, nobo.abund, mixture="NB", K=15)  
  
fmsDC <- fitList(det.null.nobo, det.weather.nobo, det.global2.nobo,  
 det.sound.nobo, det.date2.nobo, det.detect2.nobo, det.notdate.nobo,  
 det.time.nobo, det.timing2.nobo)

## Warning in fitList(det.null.nobo, det.weather.nobo, det.global2.nobo,  
## det.sound.nobo, : Your list was unnamed, so model names were added as  
## object names

msDC.nobo <- modSel(fmsDC)  
msDC.nobo

## nPars AIC delta AICwt cumltvWt  
## det.date2.nobo 5 426.91 0.00 5.8e-01 0.58  
## det.timing2.nobo 6 428.80 1.89 2.3e-01 0.81  
## det.global2.nobo 9 430.52 3.61 9.6e-02 0.91  
## det.detect2.nobo 7 430.56 3.66 9.4e-02 1.00  
## det.weather.nobo 5 448.76 21.85 1.0e-05 1.00  
## det.notdate.nobo 6 450.64 23.74 4.1e-06 1.00  
## det.null.nobo 3 463.47 36.56 6.7e-09 1.00  
## det.time.nobo 4 465.24 38.33 2.8e-09 1.00  
## det.sound.nobo 5 467.33 40.43 9.7e-10 1.00

#msDC.nobo@Full  
#summary: NB is better fit than Poisson.  
#Global is no longer best detection model.  
## date2 best model, timing2 second model.

det.date2.nobo # positive w date, - with date^2

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ 1, data = nobo.abund,   
## K = 15, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.401 0.378 1.06 0.289  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.986 0.522 -1.89 5.87e-02  
## Jdate 0.920 0.184 5.00 5.72e-07  
## I(Jdate^2) -0.428 0.146 -2.93 3.43e-03  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## -0.235 0.419 -0.561 0.575  
##   
## AIC: 426.9056

confint(det.date2.nobo, type="det",method="normal")

## 0.025 0.975  
## p(Int) -2.0079440 0.03642716  
## p(Jdate) 0.5596966 1.28126488  
## p(I(Jdate^2)) -0.7141012 -0.14117942

confint(det.date2.nobo, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.3406653 1.14235

det.timing2.nobo # positive w date, - with date^2

##   
## Call:  
## pcount(formula = ~Time + Jdate + I(Jdate^2) ~ 1, data = nobo.abund,   
## K = 15, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.404 0.381 1.06 0.289  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.9908 0.525 -1.886 5.92e-02  
## Time 0.0445 0.135 0.331 7.41e-01  
## Jdate 0.9247 0.185 5.002 5.67e-07  
## I(Jdate^2) -0.4293 0.146 -2.937 3.32e-03  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## -0.23 0.421 -0.545 0.586  
##   
## AIC: 428.7954

confint(det.timing2.nobo, type="det",method="normal")

## 0.025 0.975  
## p(Int) -2.0201259 0.03859031  
## p(Time) -0.2191537 0.30823218  
## p(Jdate) 0.5623470 1.28697849  
## p(I(Jdate^2)) -0.7158106 -0.14277119

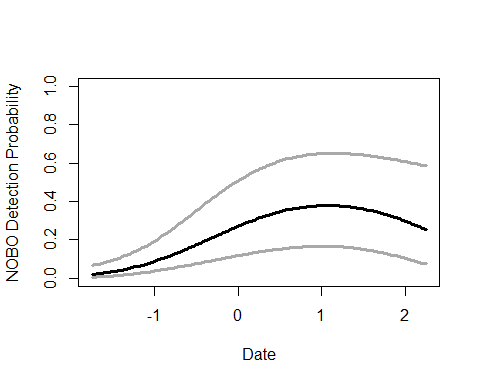
confint(det.timing2.nobo, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.3427265 1.151386

# jdate variable - date2, averaged with timing2 model too  
det.date2.nobo <- pcount(~ Jdate + I(Jdate^2) ~1, nobo.abund, mixture="NB", K=15)  
det.timing2.nobo <-pcount(~ Time + Jdate + I(Jdate^2) ~1, nobo.abund, mixture="NB", K=15)  
  
#summary(inbu.abund)  
dms\_top.nobo <- fitList(det.date2.nobo,det.timing2.nobo)

## Warning in fitList(det.date2.nobo, det.timing2.nobo): Your list was  
## unnamed, so model names were added as object names

ND.nobod <-data.frame(Jdate=seq(-1.75,2.25,length=100),Time=0)  
nobo.est.date <- predict(dms\_top.nobo, type="det",  
 newdata=ND.nobod,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=nobo.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Date", ylab="NOBO Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=nobo.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=nobo.est.date, type="l", lwd=3, col="darkgray")



write.table(msDC.nobo@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/nobo\_top\_models\_msDC.xls",sep="\t")

#yes: HW\_dens\_1050, FG\_shrub, NHW\_saplings  
#no: BA, FG\_herb  
testR.nobo <- pcount(~1 ~BA, nobo.abund, mixture="P", K=4)  
testQ.nobo <- pcount(~1 ~BA + I(BA^2), nobo.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.nobo, testQ.nobo)

## Warning in fitList(testR.nobo, testQ.nobo): Your list was unnamed, so model  
## names were added as object names

BAtest.nobo <- modSel(msBAtest)  
BAtest.nobo

## nPars AIC delta AICwt cumltvWt  
## testR.nobo 3 475.57 0.00 0.68 0.68  
## testQ.nobo 4 477.12 1.55 0.32 1.00

#yes  
testR.nobo <- pcount(~1 ~HW\_dens\_1050, nobo.abund, mixture="P", K=4)  
testQ.nobo <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), nobo.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.nobo, testQ.nobo)

## Warning in fitList(testR.nobo, testQ.nobo): Your list was unnamed, so model  
## names were added as object names

HW1050test.nobo <- modSel(msHW1050test)  
HW1050test.nobo

## nPars AIC delta AICwt cumltvWt  
## testQ.nobo 4 471.51 0.00 0.9977 1.00  
## testR.nobo 3 483.64 12.13 0.0023 1.00

testR.nobo <- pcount(~1 ~FG\_herb, nobo.abund, mixture="P", K=4)  
testQ.nobo <- pcount(~1 ~FG\_herb + I(FG\_herb^2), nobo.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.nobo, testQ.nobo)

## Warning in fitList(testR.nobo, testQ.nobo): Your list was unnamed, so model  
## names were added as object names

FGHtest.nobo <- modSel(msFGHtest)  
FGHtest.nobo

## nPars AIC delta AICwt cumltvWt  
## testQ.nobo 4 454.57 0.00 0.936 0.94  
## testR.nobo 3 459.92 5.35 0.064 1.00

testR.nobo <- pcount(~1 ~FG\_shrub, nobo.abund, mixture="P", K=4)  
testQ.nobo <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), nobo.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.nobo, testQ.nobo)

## Warning in fitList(testR.nobo, testQ.nobo): Your list was unnamed, so model  
## names were added as object names

FGStest.nobo <- modSel(msFGStest)  
FGStest.nobo

## nPars AIC delta AICwt cumltvWt  
## testR.nobo 3 479.50 0.00 0.55 0.55  
## testQ.nobo 4 479.91 0.41 0.45 1.00

testR.nobo <- pcount(~1 ~NHW\_saplings, nobo.abund, mixture="P", K=4)  
testQ.nobo <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), nobo.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.nobo, testQ.nobo)

## Warning in fitList(testR.nobo, testQ.nobo): Your list was unnamed, so model  
## names were added as object names

HWStest.nobo <- modSel(msHWStest)  
HWStest.nobo

## nPars AIC delta AICwt cumltvWt  
## testR.nobo 3 486.18 0.00 0.73 0.73  
## testQ.nobo 4 488.18 2.00 0.27 1.00

#more appropriate detection covariates (now, date2 not global)  
null.nobo <- pcount(~ Jdate + I(Jdate^2) ~1, nobo.abund, mixture="NB", K=80)  
global.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + Herbicide + BA +Ccover + TreeHt  
 + Ldepth + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + I(HW\_dens\_1050^2)  
 + FG\_herb+ I(FG\_herb^2) + FG\_shrub  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , nobo.abund, mixture="NB", K=80)  
#FPSiteIndex taken out, TreeHt, NHW\_saplings, NP\_over\_20cm, FG\_herb, Rel\_HW2P\_canopy, Nsnags  
local.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , nobo.abund, mixture="NB", K=80) #can only include BA OR CCover  
lh.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ BA + Ldepth + HW\_dens\_1050 + I(HW\_dens\_1050^2) + FG\_shrub + Age + YearCat  
 , nobo.abund, mixture="NB", K=80)  
landmetrics.nobo <- pcount (~ Jdate + I(Jdate^2)  
 ~ Parea + ShapeIndex + YearCat  
 , nobo.abund, mixture="NB",K=80)  
landscape500.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass500m + HighDev500m + Schrubs500m + Evergreen500m  
 + Ag500m + YearCat  
 , nobo.abund, mixture="NB", K=80)  
landscape1.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass1km + HighDev1km + Schrubs1km + Evergreen1km + YearCat  
 , nobo.abund, mixture="NB", K=80)  
landscape5.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass5km + HighDev5km + Schrubs5km + Evergreen5km + YearCat  
 , nobo.abund, mixture="NB", K=80)  
landscape30.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass30km + HighDev30km + Evergreen30km + YearCat  
 , nobo.abund, mixture="NB", K=80) #removed Protected  
treatment.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Nburns + Nthins + YearCat  
 , nobo.abund, mixture ="NB", K=80) #this one I made Nburns instead of treatment for ones with many many burns  
management.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , nobo.abund, mixture="NB", K=80)  
disturbance.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , nobo.abund, mixture="NB", K=80)  
siteprod.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ PISoils + NSoilTypes + YearCat  
 , nobo.abund, mixture="NB", K=80) #no FPSiteIndex  
#upstate.nobo <- pcount(~ Jdate + Wind + Sky + Noise +Time ~ X + Y + Z, nobo.abund, mixture="NB", K=40)  
coord.nobo <- pcount (~Jdate + I(Jdate^2)  
 ~ Latitude + Longitude + YearCat  
 ,nobo.abund, mixture="P", K=80)  
  
fms <- fitList(null.nobo, local.nobo, lh.nobo, landmetrics.nobo,  
 landscape500.nobo, landscape1.nobo, landscape5.nobo, landscape30.nobo,  
 treatment.nobo, management.nobo, disturbance.nobo,  
 siteprod.nobo, coord.nobo)

## Warning in fitList(null.nobo, local.nobo, lh.nobo, landmetrics.nobo,  
## landscape500.nobo, : Your list was unnamed, so model names were added as  
## object names

ms.nobo <- modSel(fms) #note this does not include FPSiteIndex or upstate OR GLOBAL  
ms.nobo

## nPars AIC delta AICwt cumltvWt  
## landscape1.nobo 10 388.67 0.00 1.0e+00 1.00  
## landscape30.nobo 9 404.16 15.49 4.3e-04 1.00  
## landscape5.nobo 10 405.29 16.62 2.5e-04 1.00  
## landscape500.nobo 11 409.43 20.76 3.1e-05 1.00  
## lh.nobo 12 412.89 24.23 5.5e-06 1.00  
## treatment.nobo 8 414.88 26.22 2.0e-06 1.00  
## disturbance.nobo 8 415.45 26.78 1.5e-06 1.00  
## coord.nobo 7 416.54 27.88 8.8e-07 1.00  
## local.nobo 9 417.81 29.14 4.7e-07 1.00  
## management.nobo 13 418.46 29.79 3.4e-07 1.00  
## landmetrics.nobo 8 424.43 35.77 1.7e-08 1.00  
## null.nobo 5 426.90 38.24 5.0e-09 1.00  
## siteprod.nobo 8 428.78 40.11 1.9e-09 1.00

#ms.nobo@Full

landscape1.nobo

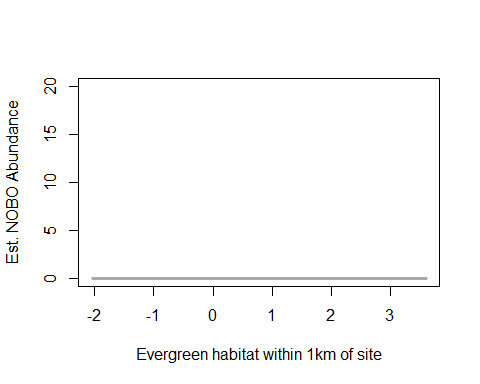
##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) ~ Grass1km + HighDev1km +   
## Schrubs1km + Evergreen1km + YearCat, data = nobo.abund, K = 80,   
## mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) -28.2084 91.596 -0.308 0.758109  
## Grass1km 0.1058 0.155 0.683 0.494882  
## HighDev1km -125.7050 409.087 -0.307 0.758629  
## Schrubs1km 0.0642 0.136 0.473 0.636123  
## Evergreen1km 0.7429 0.208 3.573 0.000352  
## YearCatB 0.6312 0.275 2.297 0.021608  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.080 0.595 -1.82 6.95e-02  
## Jdate 0.897 0.186 4.82 1.40e-06  
## I(Jdate^2) -0.412 0.148 -2.79 5.31e-03  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## 1.33 0.981 1.35 0.177  
##   
## AIC: 388.6659

confint(landscape1.nobo, type="state",method="normal")

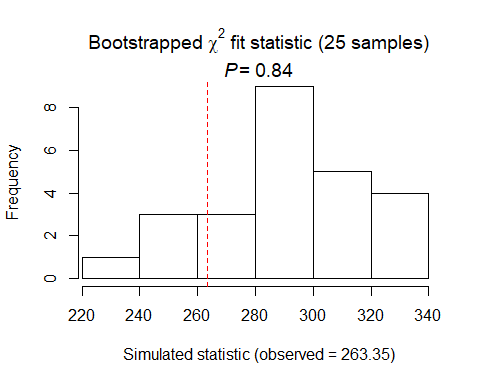
## 0.025 0.975  
## lam(Int) -207.73381989 151.3169582  
## lam(Grass1km) -0.19791209 0.4094165  
## lam(HighDev1km) -927.50104284 676.0910122  
## lam(Schrubs1km) -0.20184572 0.3303043  
## lam(Evergreen1km) 0.33544028 1.1504016  
## lam(YearCatB) 0.09266283 1.1697856

write.table(ms.nobo@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/nobo\_top\_models\_ms.xls",sep="\t")

#landscape 1 model - evergreen1km is + but this figure isn't working....  
landscape1.nobo <- pcount(~ Jdate + I(Jdate^2)  
 ~ Grass1km + HighDev1km + Schrubs1km + Evergreen1km + YearCat  
 , nobo.abund, mixture="NB", K=80)  
  
ND.nobo <-data.frame(Evergreen1km=seq(min(sc$Evergreen1km),max(sc$Evergreen1km),length=100),Schrubs1km=0,Grass1km=0,HighDev1km=0,YearCat=0)  
nobo.est.evergreen <- predict(landscape1.nobo, type="state",  
newdata=ND.nobo,appendData=TRUE, mixture="NB")  
  
plot(Predicted~ Evergreen1km, data=nobo.est.evergreen, ylim=c(0,20), type="l", lwd=3,  
xlab="Evergreen habitat within 1km of site", ylab="Est. NOBO Abundance")  
##95% confidence intervals  
lines(lower~ Evergreen1km, data=nobo.est.evergreen, type="l", lwd=3, col="darkgray")  
lines(upper~ Evergreen1km, data=nobo.est.evergreen, type="l", lwd=3, col="darkgray")

 #what’s up with this figure?

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 263.3544   
## Number of bootstrap samples = 25  
## P-value = 0.84  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 223 273 293 305 339   
##   
## Estimate of c-hat = 0.91

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 263.3544

NOBO summary: NB distribution DCs: date2 () best, timing2 second model () date2 best & timing2 second best (date + both) SCs using global: Landscape1km only best model (+ Evergreen1km, +YearB)

# PIWA

# (forest, insects, tree nesting, bark forager, nest high in pines)

# covariates: HW2P ratio, canopy, tree ht, BA,

piwa.abund<- csvToUMF("piwa\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
##type may need to change for occupancy (occuRN, pcountOpen, or whichever used) ##  
#summary(piwa.abund)  
#str(piwa.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(piwa.abund)= scale (obsCovs(piwa.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(piwa.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(piwa.abund) <- sc

#run this when have CSV with both years  
null.piwa <- pcount(~Jdate + Noise + Time ~1, piwa.abund, mixture="P", K=40)  
year.piwa <- pcount(~Jdate + Noise + Time ~ YearCat, piwa.abund, mixture="P", K=40)  
fms.year.piwa<- fitList(null.piwa, year.piwa)

## Warning in fitList(null.piwa, year.piwa): Your list was unnamed, so model  
## names were added as object names

year.ms.piwa<-modSel(fms.year.piwa)  
year.ms.piwa

## nPars AIC delta AICwt cumltvWt  
## null.piwa 5 808.32 0.00 0.72 0.72  
## year.piwa 6 810.20 1.87 0.28 1.00

^ null best model but year <2

det.date.piwa <- pcount(~ Jdate ~1, piwa.abund, mixture="P", K=15)  
det.date2.piwa <- pcount(~ Jdate + I(Jdate^2) ~1, piwa.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.piwa, det.date2.piwa)

## Warning in fitList(det.date.piwa, det.date2.piwa): Your list was unnamed,  
## so model names were added as object names

DATEtest.piwa <- modSel(mstestDATE)  
DATEtest.piwa

## nPars AIC delta AICwt cumltvWt  
## det.date2.piwa 4 802.62 0.00 0.80 0.80  
## det.date.piwa 3 805.40 2.78 0.20 1.00

det.time.piwa <-pcount(~ Time ~1, piwa.abund, mixture="P",K=15)  
det.time2.piwa <-pcount(~ Time + I(Time^2) ~1, piwa.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.piwa, det.time2.piwa)

## Warning in fitList(det.time.piwa, det.time2.piwa): Your list was unnamed,  
## so model names were added as object names

TIMEtest.piwa <- modSel(mstestTIME)  
TIMEtest.piwa

## nPars AIC delta AICwt cumltvWt  
## det.time2.piwa 4 812.69 0.00 0.85 0.85  
## det.time.piwa 3 816.22 3.53 0.15 1.00

#detection covariates first  
det.null.piwa <- pcount(~1 ~1, piwa.abund, mixture="P", K=15)  
det.weather.piwa <- pcount(~ Wind + Sky ~1, piwa.abund, mixture="P", K=15)  
det.global2.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
det.sound.piwa <- pcount(~ Noise + Wind ~1, piwa.abund, mixture="P", K=15)  
det.date2.piwa <- pcount(~ Jdate + I(Jdate^2) ~1, piwa.abund, mixture="P", K=15)  
det.detect2.piwa <- pcount(~ Jdate+ I(Jdate^2) + Noise + Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
det.notdate.piwa <-pcount(~ Wind + Sky + Noise ~1, piwa.abund, mixture="P", K=15)  
det.time2.piwa <-pcount(~ Time + I(Time^2) ~1, piwa.abund, mixture="P",K=15)  
det.timing2.piwa <-pcount(~ Time + I(Time^2) + Jdate+ I(Jdate^2) ~1, piwa.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.piwa, det.weather.piwa, det.global2.piwa,  
 det.sound.piwa, det.date2.piwa, det.detect2.piwa, det.notdate.piwa,  
 det.time2.piwa, det.timing2.piwa)

## Warning in fitList(det.null.piwa, det.weather.piwa, det.global2.piwa,  
## det.sound.piwa, : Your list was unnamed, so model names were added as  
## object names

msDC.piwa <- modSel(fmsDC)  
msDC.piwa

## nPars AIC delta AICwt cumltvWt  
## det.global2.piwa 9 800.48 0.000 0.36721 0.37  
## det.timing2.piwa 6 800.49 0.014 0.36458 0.73  
## det.detect2.piwa 7 802.41 1.938 0.13932 0.87  
## det.date2.piwa 4 802.62 2.143 0.12574 1.00  
## det.weather.piwa 4 812.44 11.962 0.00093 1.00  
## det.time2.piwa 4 812.69 12.209 0.00082 1.00  
## det.sound.piwa 4 813.30 12.819 0.00060 1.00  
## det.notdate.piwa 5 813.99 13.514 0.00043 1.00  
## det.null.piwa 2 814.28 13.804 0.00037 1.00

#msDC.piwa@Full  
#summary: global best, date second, timing is third  
#changed with quadratics: global, timing2, then detect2

det.global2.piwa

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Wind + Sky + Noise + Time +   
## I(Time^2) ~ 1, data = piwa.abund, K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.964 0.133 7.25 4.02e-13  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.1919 0.2584 -0.743 0.45771  
## Jdate -0.2922 0.0893 -3.274 0.00106  
## I(Jdate^2) -0.1932 0.0850 -2.275 0.02292  
## Wind 0.0677 0.0966 0.701 0.48350  
## Sky -0.2002 0.0867 -2.310 0.02090  
## Noise 0.0117 0.0910 0.128 0.89812  
## Time -0.1973 0.0855 -2.307 0.02104  
## I(Time^2) 0.1845 0.0893 2.066 0.03879  
##   
## AIC: 800.4762

confint(det.global2.piwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.698378870 0.31457529  
## p(Jdate) -0.467154549 -0.11725265  
## p(I(Jdate^2)) -0.359735824 -0.02673581  
## p(Wind) -0.121660236 0.25704590  
## p(Sky) -0.370052146 -0.03031416  
## p(Noise) -0.166757358 0.19006734  
## p(Time) -0.364831525 -0.02969039  
## p(I(Time^2)) 0.009501195 0.35948646

confint(det.global2.piwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.7034497 1.224217

global (- with Date & date^2, - with Sky, -time+time^2)

det.timing2.piwa

##   
## Call:  
## pcount(formula = ~Time + I(Time^2) + Jdate + I(Jdate^2) ~ 1,   
## data = piwa.abund, K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.993 0.137 7.24 4.4e-13  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.245 0.2605 -0.939 0.34797  
## Time -0.138 0.0795 -1.738 0.08215  
## I(Time^2) 0.187 0.0859 2.174 0.02974  
## Jdate -0.232 0.0781 -2.972 0.00296  
## I(Jdate^2) -0.191 0.0796 -2.401 0.01636  
##   
## AIC: 800.4906

confint(det.timing2.piwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.75519291 0.26612745  
## p(Time) -0.29411968 0.01762457  
## p(I(Time^2)) 0.01834475 0.35499157  
## p(Jdate) -0.38504041 -0.07902006  
## p(I(Jdate^2)) -0.34699037 -0.03508031

confint(det.timing2.piwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.7243919 1.26192

^ timing model: +time^2, -date & date^2

det.detect2.piwa

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Noise + Time + I(Time^2) ~   
## 1, data = piwa.abund, K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.993 0.138 7.22 5.39e-13  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.2469 0.2615 -0.944 0.34500  
## Jdate -0.2351 0.0789 -2.981 0.00288  
## I(Jdate^2) -0.1921 0.0797 -2.408 0.01602  
## Noise 0.0245 0.0887 0.277 0.78193  
## Time -0.1405 0.0800 -1.755 0.07927  
## I(Time^2) 0.1897 0.0866 2.190 0.02851  
##   
## AIC: 802.4145

confint(det.detect2.piwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.7594043 0.26556233  
## p(Jdate) -0.3897562 -0.08051585  
## p(I(Jdate^2)) -0.3483504 -0.03576502  
## p(Noise) -0.1492296 0.19831299  
## p(Time) -0.2973570 0.01640976  
## p(I(Time^2)) 0.0199477 0.35948391

confint(det.detect2.piwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.7233663 1.262944

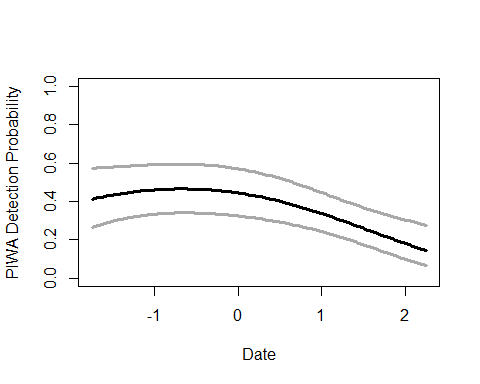
^ detect2 model: –date&date^2, +time^2

write.table(msDC.piwa@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/PIWA\_top\_models\_msDC.xls",sep="\t")

# jdate variable - averaged with global2, timing2, detect2 model  
det.global2.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
det.detect2.piwa <- pcount(~ Jdate+ I(Jdate^2) + Noise + Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
det.timing2.piwa <-pcount(~ Time + I(Time^2) + Jdate+ I(Jdate^2) ~1, piwa.abund, mixture="P", K=15)  
  
#summary(inbu.abund)  
dms\_top.piwa <- fitList(det.global2.piwa,det.timing2.piwa, det.detect2.piwa)

## Warning in fitList(det.global2.piwa, det.timing2.piwa, det.detect2.piwa):  
## Your list was unnamed, so model names were added as object names

ND.piwad <-data.frame(Jdate=seq(-1.75,2.25,length=100),Time=0, Wind=0,Sky=0,Noise=0)  
piwa.est.date <- predict(dms\_top.piwa, type="det",  
 newdata=ND.piwad,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=piwa.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Date", ylab="PIWA Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=piwa.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=piwa.est.date, type="l", lwd=3, col="darkgray")



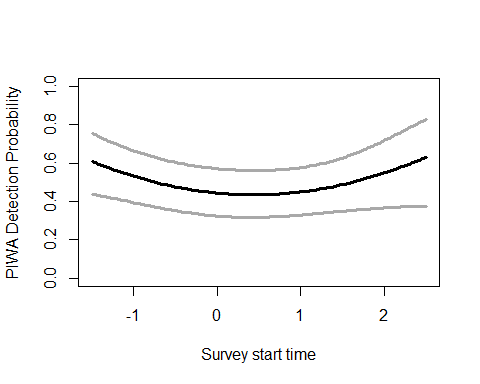
# time variable - averaged with global2, timing2, detect2 model  
det.global2.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
det.detect2.piwa <- pcount(~ Jdate+ I(Jdate^2) + Noise + Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
det.timing2.piwa <-pcount(~ Time + I(Time^2) + Jdate+ I(Jdate^2) ~1, piwa.abund, mixture="P", K=15)  
  
summary(obsCovs(piwa.abund))

## Noise Wind Sky Jdate   
## Min. :-1.0298 Min. :-1.0690 Min. :-1.17271 Min. :-1.61739   
## 1st Qu.:-1.0298 1st Qu.:-1.0690 1st Qu.:-1.17271 1st Qu.:-0.95863   
## Median : 0.1654 Median : 0.2385 Median : 0.02255 Median :-0.05061   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 1.0618 3rd Qu.: 0.2385 3rd Qu.: 1.21782 3rd Qu.: 0.80399   
## Max. : 2.5558 Max. : 2.8535 Max. : 4.80361 Max. : 2.22834   
## NA's :86 NA's :86 NA's :86 NA's :86   
## Time   
## Min. :-1.4515   
## 1st Qu.:-0.9236   
## Median :-0.1419   
## Mean : 0.0000   
## 3rd Qu.: 0.7543   
## Max. : 2.4936   
## NA's :86

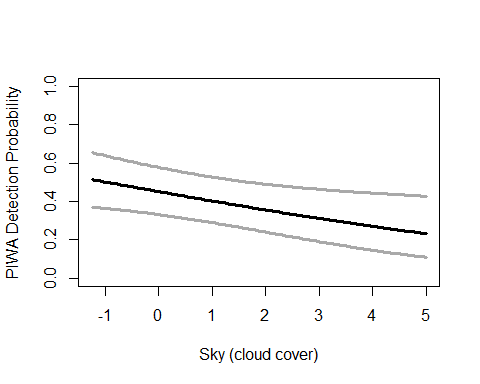
dms\_top.piwa <- fitList(det.global2.piwa,det.timing2.piwa, det.detect2.piwa)

## Warning in fitList(det.global2.piwa, det.timing2.piwa, det.detect2.piwa):  
## Your list was unnamed, so model names were added as object names

ND.piwad2 <-data.frame(Time=seq(-1.5,2.5,length=100),Jdate=0, Wind=0,Sky=0,Noise=0)  
piwa.est.time <- predict(dms\_top.piwa, type="det",  
 newdata=ND.piwad2,appendData=TRUE)  
  
plot(Predicted~ Time, data=piwa.est.time, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey start time", ylab="PIWA Detection Probability")  
##95% confidence intervals  
lines(lower~ Time, data=piwa.est.time, type="l", lwd=3, col="darkgray")  
lines(upper~ Time, data=piwa.est.time, type="l", lwd=3, col="darkgray")



# sky variable - only global  
det.global2.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2) ~1, piwa.abund, mixture="P", K=15)  
  
#summary(obsCovs(piwa.abund))  
  
ND.piwad3 <-data.frame(Sky=seq(-1.25,5,length=100),Jdate=0, Wind=0,Time=0,Noise=0)  
piwa.est.sky <- predict(det.global2.piwa, type="det",  
 newdata=ND.piwad3,appendData=TRUE)  
  
plot(Predicted~ Sky, data=piwa.est.sky, ylim=c(0,1), type="l", lwd=3,  
 xlab="Sky (cloud cover)", ylab="PIWA Detection Probability")  
##95% confidence intervals  
lines(lower~ Sky, data=piwa.est.sky, type="l", lwd=3, col="darkgray")  
lines(upper~ Sky, data=piwa.est.sky, type="l", lwd=3, col="darkgray")



#yes: BA, FG\_shrub -- no to others  
testR.piwa <- pcount(~1 ~BA, piwa.abund, mixture="P", K=20)  
testQ.piwa <- pcount(~1 ~BA + I(BA^2), piwa.abund, mixture="P", K=20)  
msBAtest <- fitList(testR.piwa, testQ.piwa)

## Warning in fitList(testR.piwa, testQ.piwa): Your list was unnamed, so model  
## names were added as object names

BAtest.piwa <- modSel(msBAtest)  
BAtest.piwa

## nPars AIC delta AICwt cumltvWt  
## testQ.piwa 4 808.37 0.00 0.87 0.87  
## testR.piwa 3 812.10 3.72 0.13 1.00

testR.piwa <- pcount(~1 ~HW\_dens\_1050, piwa.abund, mixture="P", K=20)  
testQ.piwa <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), piwa.abund, mixture="P", K=20)  
msHW1050test <- fitList(testR.piwa, testQ.piwa)

## Warning in fitList(testR.piwa, testQ.piwa): Your list was unnamed, so model  
## names were added as object names

HW1050test.piwa <- modSel(msHW1050test)  
HW1050test.piwa

## nPars AIC delta AICwt cumltvWt  
## testR.piwa 3 814.61 0.00 0.71 0.71  
## testQ.piwa 4 816.39 1.78 0.29 1.00

testR.piwa <- pcount(~1 ~FG\_herb, piwa.abund, mixture="P", K=20)  
testQ.piwa <- pcount(~1 ~FG\_herb + I(FG\_herb^2), piwa.abund, mixture="P", K=20)  
msFGHtest <- fitList(testR.piwa, testQ.piwa)

## Warning in fitList(testR.piwa, testQ.piwa): Your list was unnamed, so model  
## names were added as object names

FGHtest.piwa <- modSel(msFGHtest)  
FGHtest.piwa

## nPars AIC delta AICwt cumltvWt  
## testR.piwa 3 813.67 0.00 0.72 0.72  
## testQ.piwa 4 815.56 1.89 0.28 1.00

testR.piwa <- pcount(~1 ~FG\_shrub, piwa.abund, mixture="P", K=20)  
testQ.piwa <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), piwa.abund, mixture="P", K=20)  
msFGStest <- fitList(testR.piwa, testQ.piwa)

## Warning in fitList(testR.piwa, testQ.piwa): Your list was unnamed, so model  
## names were added as object names

FGStest.piwa <- modSel(msFGStest)  
FGStest.piwa

## nPars AIC delta AICwt cumltvWt  
## testQ.piwa 4 813.43 0.00 0.68 0.68  
## testR.piwa 3 814.90 1.47 0.32 1.00

testR.piwa <- pcount(~1 ~NHW\_saplings, piwa.abund, mixture="P", K=20)  
testQ.piwa <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), piwa.abund, mixture="P", K=20)  
msHWStest <- fitList(testR.piwa, testQ.piwa)

## Warning in fitList(testR.piwa, testQ.piwa): Your list was unnamed, so model  
## names were added as object names

HWStest.piwa <- modSel(msHWStest)  
HWStest.piwa

## nPars AIC delta AICwt cumltvWt  
## testR.piwa 3 815.37 0.00 0.71 0.71  
## testQ.piwa 4 817.15 1.78 0.29 1.00

#more appropriate detection covariates (global2 now)  
null.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~1, piwa.abund, mixture="P", K=40)  
global.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Treatment + Herbicide + BA + I(BA^2) + Ccover  
 + Ldepth + TreeHt + TimeSinceB + TimeSinceT + Nthins  
 + NP\_over\_20cm  
 + Rel\_HW2P\_canopy + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , piwa.abund, mixture="P", K=40) #FPSiteIndex taken out, Age, HWs, snags  
local.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , piwa.abund, mixture="P", K=40) #can only include BA OR CCover  
lh.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Ccover + TreeHt + Rel\_HW2P\_canopy + NP\_over\_20cm + YearCat  
 , piwa.abund, mixture="P", K=40)  
landmetrics.piwa <- pcount (~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2) ~ Parea + ShapeIndex + YearCat  
 , piwa.abund, mixture="P",K=40)  
landscape500.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Evergreen500m + HighDev500m + YearCat  
 , piwa.abund, mixture="P", K=40)  
landscape1.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Evergreen1km + HighDev1km + YearCat  
 , piwa.abund, mixture="P", K=40)  
landscape5.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Evergreen5km + HighDev5km + YearCat  
 , piwa.abund, mixture="P", K=40)  
landscape30.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Evergreen30km + HighDev30km + YearCat  
 , piwa.abund, mixture="P", K=40) #removed Protected  
treatment.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Treatment + Nthins + YearCat  
 , piwa.abund, mixture ="P", K=40)  
management.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Treatment + BA + I(BA^2) + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , piwa.abund, mixture="P", K=40)  
disturbance.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , piwa.abund, mixture="P", K=40)  
siteprod.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ PISoils + NSoilTypes + YearCat  
 , piwa.abund, mixture="P", K=40) #Site Index out  
#upstate.piwa <- pcount(~ Jdate + Time ~ X + Y + Z, piwa.abund, mixture="P", K=40)  
coord.piwa<- pcount (~Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)~ Latitude + Longitude + YearCat  
 , piwa.abund, mixture="P", K=40)  
  
  
fms <- fitList(null.piwa, global.piwa, local.piwa, lh.piwa, landmetrics.piwa,  
 landscape500.piwa, landscape1.piwa, landscape5.piwa, landscape30.piwa,  
 treatment.piwa, management.piwa, disturbance.piwa,  
 siteprod.piwa, coord.piwa)

## Warning in fitList(null.piwa, global.piwa, local.piwa, lh.piwa,  
## landmetrics.piwa, : Your list was unnamed, so model names were added as  
## object names

ms.piwa <- modSel(fms) #note this does not include upstate  
ms.piwa

## nPars AIC delta AICwt cumltvWt  
## coord.piwa 12 798.10 0.00 3.7e-01 0.37  
## landscape30.piwa 12 799.23 1.13 2.1e-01 0.57  
## landscape5.piwa 12 799.99 1.88 1.4e-01 0.72  
## null.piwa 9 800.48 2.38 1.1e-01 0.83  
## siteprod.piwa 12 801.61 3.51 6.3e-02 0.89  
## lh.piwa 14 803.13 5.03 3.0e-02 0.92  
## local.piwa 13 804.28 6.18 1.7e-02 0.94  
## landscape500.piwa 12 804.42 6.32 1.6e-02 0.95  
## management.piwa 18 804.59 6.49 1.4e-02 0.97  
## landmetrics.piwa 12 804.70 6.60 1.4e-02 0.98  
## landscape1.piwa 12 805.32 7.22 9.9e-03 0.99  
## disturbance.piwa 12 806.15 8.05 6.5e-03 1.00  
## treatment.piwa 14 807.60 9.50 3.2e-03 1.00  
## global.piwa 28 819.17 21.07 9.7e-06 1.00

#ms.piwa@Full  
#changed order!  
#summary: coord model, then landscape30, landscape5

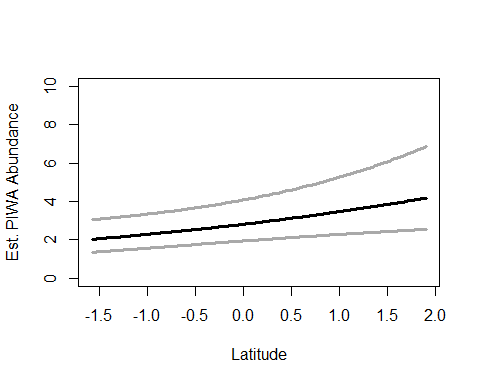
coord.piwa #+ with latitude

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Wind + Sky + Noise + Time +   
## I(Time^2) ~ Latitude + Longitude + YearCat, data = piwa.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.0371 0.1888 5.493 3.95e-08  
## Latitude 0.2074 0.0751 2.760 5.78e-03  
## Longitude -0.0255 0.0691 -0.369 7.12e-01  
## YearCatB 0.0526 0.1459 0.361 7.18e-01  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.3951 0.3174 -1.245 0.21319  
## Jdate -0.2524 0.0871 -2.897 0.00377  
## I(Jdate^2) -0.1805 0.0837 -2.157 0.03098  
## Wind 0.0716 0.0943 0.759 0.44785  
## Sky -0.1791 0.0836 -2.143 0.03214  
## Noise -0.0257 0.0866 -0.297 0.76619  
## Time -0.1728 0.0817 -2.115 0.03447  
## I(Time^2) 0.1611 0.0837 1.925 0.05426  
##   
## AIC: 798.1011

confint(coord.piwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.66705962 1.4071478  
## lam(Latitude) 0.06011466 0.3545984  
## lam(Longitude) -0.16092531 0.1099576  
## lam(YearCatB) -0.23324773 0.3385274

#coords with latitude variable  
coord.piwa<- pcount (~Jdate + I(Jdate^2) + Wind + Sky + Noise  
 +Time + I(Time^2)~ Latitude + Longitude + YearCat  
 , piwa.abund, mixture="P", K=40)  
  
  
ND.piwa <-data.frame(Latitude=seq(min(sc$Latitude),max(sc$Latitude),length=100),Longitude=0,YearCat=0)  
piwa.est.lat <- predict(coord.piwa, type="state",  
newdata=ND.piwa,appendData=TRUE)  
  
plot(Predicted~ Latitude, data=piwa.est.lat, ylim=c(0,10), type="l", lwd=3,  
xlab="Latitude", ylab="Est. PIWA Abundance")  
##95% confidence intervals  
lines(lower~ Latitude, data=piwa.est.lat, type="l", lwd=3, col="darkgray")  
lines(upper~ Latitude, data=piwa.est.lat, type="l", lwd=3, col="darkgray")



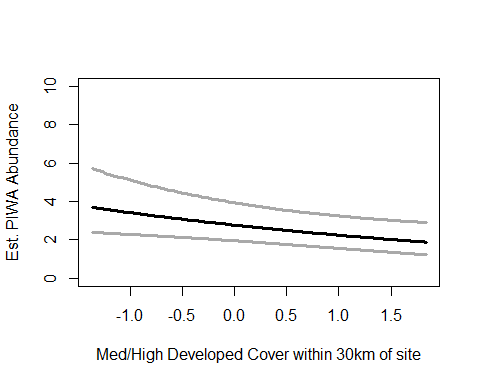
landscape30.piwa #- with highdev30

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Wind + Sky + Noise + Time +   
## I(Time^2) ~ Evergreen30km + HighDev30km + YearCat, data = piwa.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.0208 0.1781 5.731 9.98e-09  
## Evergreen30km 0.0227 0.0728 0.312 7.55e-01  
## HighDev30km -0.2095 0.0830 -2.523 1.16e-02  
## YearCatB 0.0358 0.1472 0.243 8.08e-01  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.355 0.3045 -1.165 0.24395  
## Jdate -0.254 0.0879 -2.892 0.00382  
## I(Jdate^2) -0.177 0.0842 -2.098 0.03593  
## Wind 0.084 0.0952 0.882 0.37751  
## Sky -0.175 0.0845 -2.071 0.03835  
## Noise -0.016 0.0875 -0.183 0.85459  
## Time -0.177 0.0826 -2.139 0.03241  
## I(Time^2) 0.164 0.0850 1.929 0.05371  
##   
## AIC: 799.2342

confint(landscape30.piwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.6716943 1.36990863  
## lam(Evergreen30km) -0.1200793 0.16548058  
## lam(HighDev30km) -0.3722989 -0.04678644  
## lam(YearCatB) -0.2527320 0.32443123

# high dev variable with landscape 30km  
landscape30.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Evergreen30km + HighDev30km + YearCat  
 , piwa.abund, mixture="P", K=40) #removed Protected  
  
  
ND2.piwa <-data.frame(HighDev30km=seq(min(sc$HighDev30km),max(sc$HighDev30km),length=100),Evergreen30km=0,YearCat=0)  
piwa.est.hdev30 <- predict(landscape30.piwa, type="state", newdata=ND2.piwa,appendData=TRUE)  
  
plot(Predicted~ HighDev30km, data=piwa.est.hdev30, ylim=c(0,10), type="l", lwd=3,  
xlab="Med/High Developed Cover within 30km of site", ylab="Est. PIWA Abundance")  
##95% confidence intervals  
lines(lower~ HighDev30km, data=piwa.est.hdev30, type="l", lwd=3, col="darkgray")  
lines(upper~ HighDev30km, data=piwa.est.hdev30, type="l", lwd=3, col="darkgray")



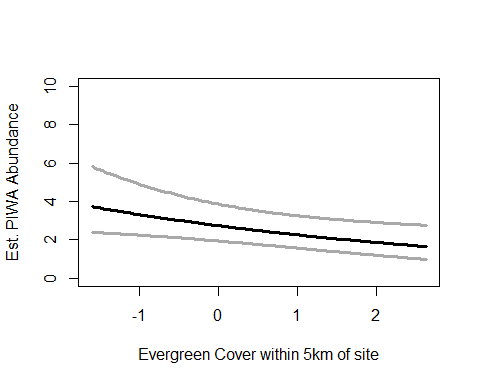
landscape5.piwa #- with evergreen

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Wind + Sky + Noise + Time +   
## I(Time^2) ~ Evergreen5km + HighDev5km + YearCat, data = piwa.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 1.00903 0.1748 5.7712 7.87e-09  
## Evergreen5km -0.19134 0.0801 -2.3895 1.69e-02  
## HighDev5km -0.00648 0.0739 -0.0877 9.30e-01  
## YearCatB 0.04518 0.1466 0.3081 7.58e-01  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.3346 0.2979 -1.123 0.26128  
## Jdate -0.2598 0.0879 -2.957 0.00311  
## I(Jdate^2) -0.1784 0.0843 -2.117 0.03428  
## Wind 0.0769 0.0958 0.802 0.42247  
## Sky -0.1799 0.0849 -2.118 0.03414  
## Noise -0.0157 0.0888 -0.177 0.85924  
## Time -0.1758 0.0831 -2.114 0.03449  
## I(Time^2) 0.1612 0.0853 1.890 0.05876  
##   
## AIC: 799.9853

confint(landscape5.piwa, type="state",method="normal")

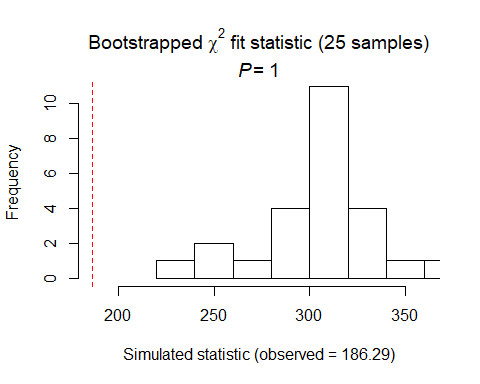
## 0.025 0.975  
## lam(Int) 0.6663537 1.3517064  
## lam(Evergreen5km) -0.3482816 -0.0343948  
## lam(HighDev5km) -0.1512426 0.1382835  
## lam(YearCatB) -0.2422251 0.3325907

# evergreen variable with landscape 5km  
landscape5.piwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise +Time + I(Time^2)  
 ~ Evergreen5km + HighDev5km + YearCat  
 , piwa.abund, mixture="P", K=40)  
  
ND3.piwa <-data.frame(Evergreen5km=seq(min(sc$Evergreen5km),max(sc$Evergreen5km),length=100),HighDev5km=0,YearCat=0)  
piwa.est.evergreen5 <- predict(landscape5.piwa, type="state", newdata=ND3.piwa,appendData=TRUE)  
  
plot(Predicted~ Evergreen5km, data=piwa.est.evergreen5, ylim=c(0,10), type="l", lwd=3,  
xlab="Evergreen Cover within 5km of site", ylab="Est. PIWA Abundance")  
##95% confidence intervals  
lines(lower~ Evergreen5km, data=piwa.est.evergreen5, type="l", lwd=3, col="darkgray")  
lines(upper~ Evergreen5km, data=piwa.est.evergreen5, type="l", lwd=3, col="darkgray")



write.table(ms.piwa@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/piwa\_top\_models\_ms.xls",sep="\t")

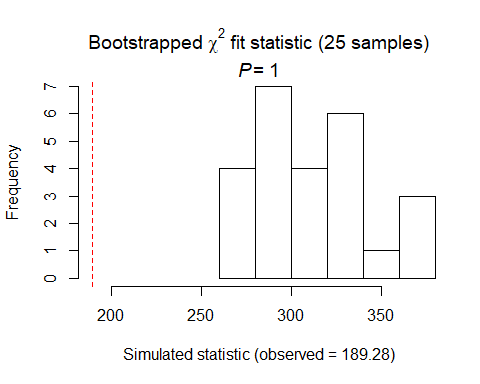
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 186.2876   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 225 298 306 318 361   
##   
## Estimate of c-hat = 0.61

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 186.2876

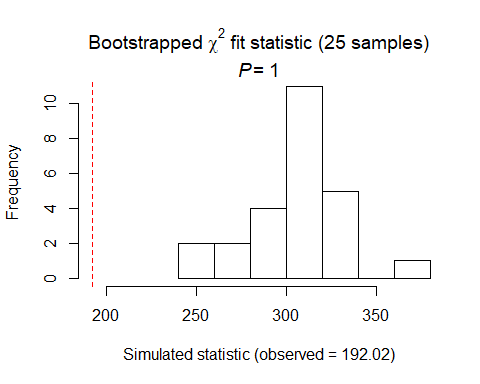
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 189.2808   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 264 289 311 336 375   
##   
## Estimate of c-hat = 0.61

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 189.2808

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 192.0159   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 249 296 304 316 378   
##   
## Estimate of c-hat = 0.63

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 192.0159

PIWA Summary P distribution DCs: -now global (–date&date^2, - time+time^2, - with sky), then timing2 (+time^2, -date & date^2), then detect2 (same as timing) SCs with global: Coordinates (+ with latitude) Landscape30 (- with HighDev 30) Landscape5 (- with Evergreen5km)

# PRWA

# (foliage gleaner, insects, shrub/tree nester 1-45’, open wood warbler, ESS/second growth brushy/bushy habitat)

# covariates: grasses, understory growth, midstory shrub density,

prwa.abund<- csvToUMF("prwa\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(prwa.abund)  
#str(prwa.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(prwa.abund)= scale (obsCovs(prwa.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(prwa.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(prwa.abund) <- sc

#run this when have CSV with both years  
null.prwa <- pcount(~Jdate + Noise + Time ~1, prwa.abund, mixture="P", K=40)  
year.prwa <- pcount(~Jdate + Noise + Time ~ YearCat, prwa.abund, mixture="P", K=40)  
fms.year.prwa<- fitList(null.prwa, year.prwa)

## Warning in fitList(null.prwa, year.prwa): Your list was unnamed, so model  
## names were added as object names

year.ms.prwa<-modSel(fms.year.prwa)  
year.ms.prwa

## nPars AIC delta AICwt cumltvWt  
## null.prwa 5 355.75 0.00 0.68 0.68  
## year.prwa 6 357.26 1.52 0.32 1.00

^ null ranked higher but year d1.52

#test for NB or Poisson - most should use Poisson  
testP.prwa <- pcount(~1 ~1, prwa.abund, mixture="P", K=4)  
testNB.prwa <- pcount(~1 ~1, prwa.abund, mixture="NB", K=4)  
fmsTEST <- fitList(testP.prwa, testNB.prwa)

## Warning in fitList(testP.prwa, testNB.prwa): Your list was unnamed, so  
## model names were added as object names

msTEST.prwa <- modSel(fmsTEST)  
msTEST.prwa

## nPars AIC delta AICwt cumltvWt  
## testNB.prwa 3 359.20 0.00 0.78 0.78  
## testP.prwa 2 361.75 2.55 0.22 1.00

#NB is best for prwa. Changed below to correspond!

det.date.prwa <- pcount(~ Jdate ~1, prwa.abund, mixture="NB", K=15)  
det.date2.prwa <- pcount(~ Jdate + I(Jdate^2) ~1, prwa.abund, mixture="NB", K=15)  
mstestDATE <- fitList(det.date.prwa, det.date2.prwa)

## Warning in fitList(det.date.prwa, det.date2.prwa): Your list was unnamed,  
## so model names were added as object names

DATEtest.prwa <- modSel(mstestDATE)  
DATEtest.prwa

## nPars AIC delta AICwt cumltvWt  
## det.date2.prwa 5 356.39 0.00 0.72 0.72  
## det.date.prwa 4 358.31 1.91 0.28 1.00

det.time.prwa <-pcount(~ Time ~1, prwa.abund, mixture="NB",K=15)  
det.time2.prwa <-pcount(~ Time + I(Time^2) ~1, prwa.abund, mixture="NB",K=15)  
mstestTIME <- fitList(det.time.prwa, det.time2.prwa)

## Warning in fitList(det.time.prwa, det.time2.prwa): Your list was unnamed,  
## so model names were added as object names

TIMEtest.prwa <- modSel(mstestTIME)  
TIMEtest.prwa

## nPars AIC delta AICwt cumltvWt  
## det.time.prwa 4 351.65 0.00 0.61 0.61  
## det.time2.prwa 5 352.56 0.92 0.39 1.00

#detection covariates first  
det.null.prwa <- pcount(~1 ~1, prwa.abund, mixture="NB", K=15)  
det.weather.prwa <- pcount(~ Wind + Sky ~1, prwa.abund, mixture="NB", K=15)  
det.global2.prwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise + Time ~1, prwa.abund, mixture="NB", K=15)  
det.sound.prwa <- pcount(~ Noise + Wind ~1, prwa.abund, mixture="NB", K=15)  
det.date2.prwa <- pcount(~ Jdate + I(Jdate^2) ~1, prwa.abund, mixture="NB", K=15)  
det.detect2.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time ~1, prwa.abund, mixture="NB", K=15)  
det.notdate.prwa <-pcount(~ Wind + Sky + Noise ~1, prwa.abund, mixture="NB", K=15)  
det.time.prwa <-pcount(~ Time ~1, prwa.abund, mixture="NB", K=15)  
det.timing2.prwa <-pcount(~ Time + Jdate + I(Jdate^2) ~1, prwa.abund, mixture="NB", K=15)  
  
fmsDC <- fitList(det.null.prwa, det.weather.prwa, det.global2.prwa,  
 det.sound.prwa, det.date2.prwa, det.detect2.prwa, det.notdate.prwa,  
 det.time.prwa, det.timing2.prwa)

## Warning in fitList(det.null.prwa, det.weather.prwa, det.global2.prwa,  
## det.sound.prwa, : Your list was unnamed, so model names were added as  
## object names

msDC.prwa <- modSel(fmsDC)  
msDC.prwa

## nPars AIC delta AICwt cumltvWt  
## det.detect2.prwa 7 350.56 0.00 0.3237 0.32  
## det.global2.prwa 9 351.27 0.71 0.2268 0.55  
## det.timing2.prwa 6 351.46 0.90 0.2063 0.76  
## det.time.prwa 4 351.65 1.09 0.1879 0.94  
## det.null.prwa 3 356.36 5.80 0.0178 0.96  
## det.date2.prwa 5 356.39 5.84 0.0175 0.98  
## det.sound.prwa 5 357.46 6.91 0.0102 0.99  
## det.notdate.prwa 6 358.78 8.22 0.0053 1.00  
## det.weather.prwa 5 359.13 8.57 0.0045 1.00

#msDC.prwa@Full  
#summary: first time, then detect, then timing  
# def changed: detect2, global2, timing2, then time.

det.detect2.prwa #positive with time

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Noise + Time ~ 1, data = prwa.abund,   
## K = 15, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## -0.194 0.295 -0.657 0.511  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.655 0.446 -1.469 0.1419  
## Jdate 0.107 0.158 0.676 0.4990  
## I(Jdate^2) -0.339 0.176 -1.925 0.0543  
## Noise -0.321 0.189 -1.701 0.0890  
## Time 0.445 0.175 2.539 0.0111  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## 0.0814 0.638 0.128 0.898  
##   
## AIC: 350.5576

confint(det.detect2.prwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.5294394 0.219123304  
## p(Jdate) -0.2031479 0.417099318  
## p(I(Jdate^2)) -0.6842275 0.006205038  
## p(Noise) -0.6901649 0.048902666  
## p(Time) 0.1014164 0.788450161

confint(det.detect2.prwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.7733761 0.3848602

det.global2.prwa #positive with time, - with jdate^2

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Wind + Sky + Noise + Time ~   
## 1, data = prwa.abund, K = 15, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## -0.234 0.277 -0.845 0.398  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.47982 0.444 -1.0803 0.28002  
## Jdate -0.00233 0.182 -0.0128 0.98981  
## I(Jdate^2) -0.46330 0.199 -2.3268 0.01998  
## Wind -0.37047 0.222 -1.6722 0.09449  
## Sky -0.08517 0.149 -0.5699 0.56873  
## Noise -0.25883 0.198 -1.3091 0.19050  
## Time 0.48526 0.185 2.6207 0.00878  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## 0.0491 0.626 0.0786 0.937  
##   
## AIC: 351.2692

confint(det.global2.prwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.3503789 0.39073142  
## p(Jdate) -0.3598113 0.35515253  
## p(I(Jdate^2)) -0.8535636 -0.07303791  
## p(Wind) -0.8046901 0.06375404  
## p(Sky) -0.3780867 0.20774064  
## p(Noise) -0.6463372 0.12868397  
## p(Time) 0.1223419 0.84817149

confint(det.global2.prwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.7769654 0.308886

det.timing2.prwa #positiv with time ... NOT date here

##   
## Call:  
## pcount(formula = ~Time + Jdate + I(Jdate^2) ~ 1, data = prwa.abund,   
## K = 15, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## -0.233 0.295 -0.791 0.429  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.5460 0.448 -1.218 0.2233  
## Time 0.4297 0.174 2.472 0.0134  
## Jdate 0.0823 0.157 0.525 0.5998  
## I(Jdate^2) -0.3427 0.176 -1.946 0.0517  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## -0.0145 0.61 -0.0237 0.981  
##   
## AIC: 351.459

confint(det.timing2.prwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.42475249 0.332776818  
## p(Time) 0.08900415 0.770484723  
## p(Jdate) -0.22526333 0.389933935  
## p(I(Jdate^2)) -0.68789262 0.002499076

confint(det.timing2.prwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.810731 0.3446671

det.time.prwa #positive with time

##   
## Call:  
## pcount(formula = ~Time ~ 1, data = prwa.abund, K = 15, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## -0.189 0.313 -0.605 0.545  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.909 0.414 -2.19 0.0282  
## Time 0.420 0.173 2.43 0.0151  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## -0.0938 0.587 -0.16 0.873  
##   
## AIC: 351.6459

confint(det.time.prwa, type="det",method="normal")

## 0.025 0.975  
## p(Int) -1.72059998 -0.09713502  
## p(Time) 0.08116042 0.75797750

confint(det.time.prwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.8024857 0.4239003

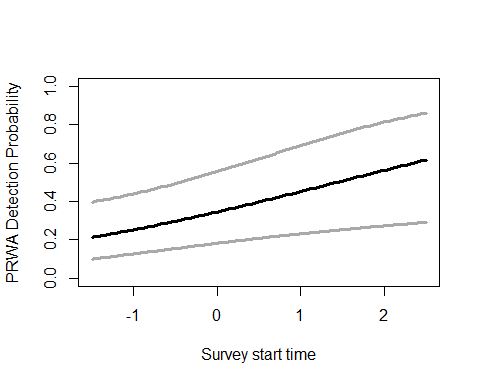
# time variable - averaged with detect2, global2, timing2, time model  
det.global2.prwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise + Time ~1, prwa.abund, mixture="NB", K=15)  
det.detect2.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time ~1, prwa.abund, mixture="NB", K=15)  
det.time.prwa <-pcount(~ Time ~1, prwa.abund, mixture="NB", K=15)  
det.timing2.prwa <-pcount(~ Time + Jdate + I(Jdate^2) ~1, prwa.abund, mixture="NB", K=15)  
  
summary(obsCovs(prwa.abund))

## Noise Wind Sky Jdate   
## Min. :-1.0298 Min. :-1.0690 Min. :-1.17271 Min. :-1.61739   
## 1st Qu.:-1.0298 1st Qu.:-1.0690 1st Qu.:-1.17271 1st Qu.:-0.95863   
## Median : 0.1654 Median : 0.2385 Median : 0.02255 Median :-0.05061   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 1.0618 3rd Qu.: 0.2385 3rd Qu.: 1.21782 3rd Qu.: 0.80399   
## Max. : 2.5558 Max. : 2.8535 Max. : 4.80361 Max. : 2.22834   
## NA's :86 NA's :86 NA's :86 NA's :86   
## Time   
## Min. :-1.4515   
## 1st Qu.:-0.9236   
## Median :-0.1419   
## Mean : 0.0000   
## 3rd Qu.: 0.7543   
## Max. : 2.4936   
## NA's :86

dms\_top.prwad <- fitList(det.global2.prwa,det.timing2.prwa, det.detect2.prwa, det.time.prwa)

## Warning in fitList(det.global2.prwa, det.timing2.prwa, det.detect2.prwa, :  
## Your list was unnamed, so model names were added as object names

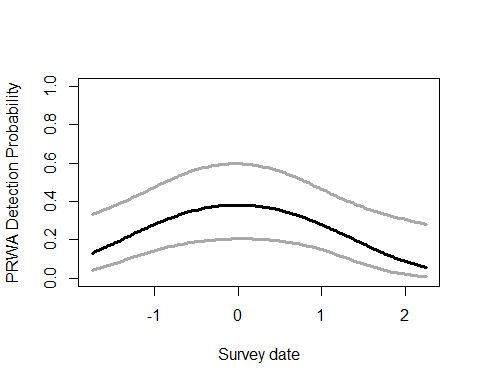
ND.prwad <-data.frame(Time=seq(-1.5,2.5,length=100),Jdate=0, Wind=0,Sky=0,Noise=0)  
prwa.est.time <- predict(dms\_top.prwad, type="det",  
 newdata=ND.prwad,appendData=TRUE)  
  
plot(Predicted~ Time, data=prwa.est.time, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey start time", ylab="PRWA Detection Probability")  
##95% confidence intervals  
lines(lower~ Time, data=prwa.est.time, type="l", lwd=3, col="darkgray")  
lines(upper~ Time, data=prwa.est.time, type="l", lwd=3, col="darkgray")



#date^2 within global model  
det.global2.prwa <- pcount(~ Jdate + I(Jdate^2) + Wind + Sky + Noise + Time ~1, prwa.abund, mixture="NB", K=15)  
  
summary(obsCovs(prwa.abund))

## Noise Wind Sky Jdate   
## Min. :-1.0298 Min. :-1.0690 Min. :-1.17271 Min. :-1.61739   
## 1st Qu.:-1.0298 1st Qu.:-1.0690 1st Qu.:-1.17271 1st Qu.:-0.95863   
## Median : 0.1654 Median : 0.2385 Median : 0.02255 Median :-0.05061   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 1.0618 3rd Qu.: 0.2385 3rd Qu.: 1.21782 3rd Qu.: 0.80399   
## Max. : 2.5558 Max. : 2.8535 Max. : 4.80361 Max. : 2.22834   
## NA's :86 NA's :86 NA's :86 NA's :86   
## Time   
## Min. :-1.4515   
## 1st Qu.:-0.9236   
## Median :-0.1419   
## Mean : 0.0000   
## 3rd Qu.: 0.7543   
## Max. : 2.4936   
## NA's :86

ND.prwad2 <-data.frame(Jdate=seq(-1.75,2.25,length=100),Time=0, Wind=0,Sky=0,Noise=0)  
prwa.est.date <- predict(det.global2.prwa, type="det",  
 newdata=ND.prwad2,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=prwa.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey date", ylab="PRWA Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=prwa.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=prwa.est.date, type="l", lwd=3, col="darkgray")



write.table(msDC.prwa@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/prwa\_top\_models\_msDC.xls",sep="\t")

#HW\_dens\_1050 is better quadratic; others not  
testR.prwa <- pcount(~1 ~BA, prwa.abund, mixture="NB", K=4)  
testQ.prwa <- pcount(~1 ~BA + I(BA^2), prwa.abund, mixture="NB", K=4)  
msBAtest <- fitList(testR.prwa, testQ.prwa)

## Warning in fitList(testR.prwa, testQ.prwa): Your list was unnamed, so model  
## names were added as object names

BAtest.prwa <- modSel(msBAtest)  
BAtest.prwa

## nPars AIC delta AICwt cumltvWt  
## testR.prwa 4 349.60 0.00 0.62 0.62  
## testQ.prwa 5 350.61 1.01 0.38 1.00

testR.prwa <- pcount(~1 ~HW\_dens\_1050, prwa.abund, mixture="NB", K=4)  
testQ.prwa <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), prwa.abund, mixture="NB", K=4)  
msHW1050test <- fitList(testR.prwa, testQ.prwa)

## Warning in fitList(testR.prwa, testQ.prwa): Your list was unnamed, so model  
## names were added as object names

HW1050test.prwa <- modSel(msHW1050test)  
HW1050test.prwa

## nPars AIC delta AICwt cumltvWt  
## testQ.prwa 5 350.37 0.00 0.73 0.73  
## testR.prwa 4 352.33 1.96 0.27 1.00

testR.prwa <- pcount(~1 ~FG\_herb, prwa.abund, mixture="NB", K=4)  
testQ.prwa <- pcount(~1 ~FG\_herb + I(FG\_herb^2), prwa.abund, mixture="NB", K=4)  
msFGHtest <- fitList(testR.prwa, testQ.prwa)

## Warning in fitList(testR.prwa, testQ.prwa): Your list was unnamed, so model  
## names were added as object names

FGHtest.prwa <- modSel(msFGHtest)  
FGHtest.prwa

## nPars AIC delta AICwt cumltvWt  
## testR.prwa 4 335.63 0.00 0.57 0.57  
## testQ.prwa 5 336.19 0.55 0.43 1.00

testR.prwa <- pcount(~1 ~FG\_shrub, prwa.abund, mixture="NB", K=4)  
testQ.prwa <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), prwa.abund, mixture="NB", K=4)  
msFGStest <- fitList(testR.prwa, testQ.prwa)

## Warning in fitList(testR.prwa, testQ.prwa): Your list was unnamed, so model  
## names were added as object names

FGStest.prwa <- modSel(msFGStest)  
FGStest.prwa

## nPars AIC delta AICwt cumltvWt  
## testR.prwa 4 359.27 0.00 0.67 0.67  
## testQ.prwa 5 360.66 1.39 0.33 1.00

testR.prwa <- pcount(~1 ~NHW\_saplings, prwa.abund, mixture="NB", K=4)  
testQ.prwa <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), prwa.abund, mixture="NB", K=4)  
msHWStest <- fitList(testR.prwa, testQ.prwa)

## Warning in fitList(testR.prwa, testQ.prwa): Your list was unnamed, so model  
## names were added as object names

HWStest.prwa <- modSel(msHWStest)  
HWStest.prwa

## nPars AIC delta AICwt cumltvWt  
## testR.prwa 4 360.13 0.00 0.73 0.73  
## testQ.prwa 5 362.12 1.99 0.27 1.00

#more appropriate detection covariates (Detect2) #NB  
null.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~1, prwa.abund, mixture="NB", K=60)  
global.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Treatment + Herbicide + BA +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + I(HW\_dens\_1050^2)  
 + FG\_herb + NHW\_saplings + NP\_over\_20cm  
 + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , prwa.abund, mixture="NB", K=60) # FPSiteIndex, Nsnags, Rel\_HW2P\_canopy   
local.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , prwa.abund, mixture="NB", K=60) #can only include BA OR CCover  
lh.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Age + FG\_herb + HW\_dens\_1050 + I(HW\_dens\_1050^2)  
 + NHW\_saplings + NP\_over\_20cm + YearCat  
 , prwa.abund, mixture="NB", K=60)  
landmetrics.prwa <- pcount (~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Parea + ShapeIndex + YearCat  
 , prwa.abund, mixture="NB",K=60)  
landscape500.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen500m + Grass500m + HighDev500m + Schrubs500m + YearCat  
 , prwa.abund, mixture="NB", K=60)  
landscape1.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen1km + Grass1km + HighDev1km + Schrubs1km + YearCat  
 , prwa.abund, mixture="NB", K=60)  
landscape5.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen5km + Grass5km + HighDev5km + Schrubs5km + YearCat  
 , prwa.abund, mixture="NB", K=60)  
landscape30.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen30km + Grass30km + HighDev30km + YearCat  
 , prwa.abund, mixture="NB", K=60) #rmoved Protected  
treatment.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Treatment + Nthins + YearCat  
 , prwa.abund, mixture ="NB", K=60)  
management.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , prwa.abund, mixture="NB", K=60)  
disturbance.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , prwa.abund, mixture="NB", K=60)  
siteprod.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ PISoils + NSoilTypes + YearCat  
 , prwa.abund, mixture="NB", K=60)  
#upstate.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time ~ X + Y + Z, prwa.abund, mixture="NB", K=40)  
coord.prwa <- pcount (~Jdate + I(Jdate^2) + Noise + Time ~ Latitude + Longitude + YearCat  
 , prwa.abund, mixture="P", K=60)  
  
  
fms <- fitList(null.prwa, global.prwa, local.prwa, lh.prwa, landmetrics.prwa,  
 landscape500.prwa, landscape1.prwa, landscape5.prwa, landscape30.prwa,  
 treatment.prwa, management.prwa, disturbance.prwa,  
 siteprod.prwa, coord.prwa)

## Warning in fitList(null.prwa, global.prwa, local.prwa, lh.prwa,  
## landmetrics.prwa, : Your list was unnamed, so model names were added as  
## object names

ms.prwa <- modSel(fms) #note this does not include UPSTATE

## Warning in sqrt(diag(vcov(x, altNames = TRUE))): NaNs produced

## Warning in sqrt(diag(vcov(x, altNames = TRUE))): NaNs produced

ms.prwa

## nPars AIC delta AICwt cumltvWt  
## landscape1.prwa 12 328.54 0.00 4.7e-01 0.47  
## landscape5.prwa 12 328.79 0.25 4.2e-01 0.89  
## lh.prwa 14 331.48 2.94 1.1e-01 1.00  
## landscape30.prwa 11 340.84 12.30 1.0e-03 1.00  
## landscape500.prwa 12 341.82 13.28 6.2e-04 1.00  
## management.prwa 15 348.63 20.09 2.1e-05 1.00  
## global.prwa 29 350.41 21.87 8.4e-06 1.00  
## null.prwa 7 350.56 22.02 7.8e-06 1.00  
## local.prwa 11 351.23 22.69 5.6e-06 1.00  
## treatment.prwa 12 351.44 22.90 5.0e-06 1.00  
## disturbance.prwa 10 353.33 24.79 2.0e-06 1.00  
## landmetrics.prwa 10 355.39 26.85 7.0e-07 1.00  
## siteprod.prwa 10 355.72 27.18 5.9e-07 1.00  
## coord.prwa 9 357.32 28.78 2.7e-07 1.00

#ms.prwa@Full  
#now landscape 1km, then 5km

landscape1.prwa #+evergreen, - highdev

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Noise + Time ~ Evergreen1km +   
## Grass1km + HighDev1km + Schrubs1km + YearCat, data = prwa.abund,   
## K = 60, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) -39.0460 0.298 -131.095 0.00000  
## Evergreen1km 0.4634 0.152 3.052 0.00227  
## Grass1km -0.2981 0.184 -1.624 0.10446  
## HighDev1km -172.3387 0.000 -Inf 0.00000  
## Schrubs1km 0.0406 0.157 0.259 0.79562  
## YearCatB 0.2191 0.287 0.763 0.44575  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.609 0.379 -1.608 0.10778  
## Jdate 0.108 0.159 0.678 0.49778  
## I(Jdate^2) -0.347 0.174 -1.999 0.04562  
## Noise -0.289 0.184 -1.567 0.11722  
## Time 0.498 0.174 2.859 0.00425

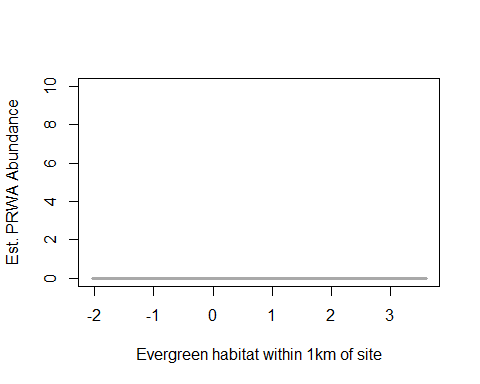
## Warning in sqrt(diag(vcov(obj))): NaNs produced

## Dispersion:  
## Estimate SE z P(>|z|)  
## 40.6 NaN NaN NaN  
##   
## AIC: 328.5413

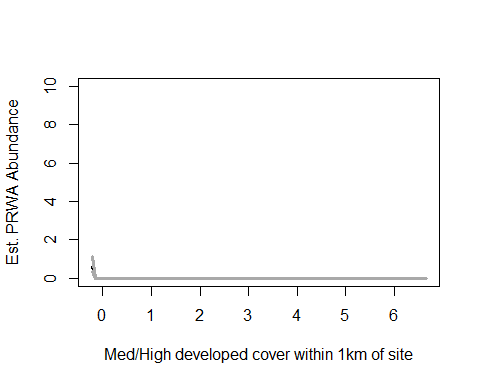
confint(landscape1.prwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -39.6297666 -38.46223649  
## lam(Evergreen1km) 0.1658139 0.76094020  
## lam(Grass1km) -0.6580506 0.06176724  
## lam(HighDev1km) -172.3387489 -172.33874892  
## lam(Schrubs1km) -0.2666321 0.34783712  
## lam(YearCatB) -0.3441001 0.78234481

#landscape1km - evergreen  
landscape1.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen1km + Grass1km + HighDev1km  
 + Schrubs1km + YearCat  
 , prwa.abund, mixture="NB", K=60)  
  
ND.prwa <-data.frame(Evergreen1km=seq(min(sc$Evergreen1km),max(sc$Evergreen1km),length=100),Grass1km=0,HighDev1km=0,Schrubs1km=0,YearCat=0)  
prwa.est.evergreen1 <- predict(landscape1.prwa, type="state",  
newdata=ND.prwa,appendData=TRUE)  
  
plot(Predicted~ Evergreen1km, data=prwa.est.evergreen1, ylim=c(0,10), type="l", lwd=3,  
xlab="Evergreen habitat within 1km of site", ylab="Est. PRWA Abundance")  
##95% confidence intervals  
lines(lower~ Evergreen1km, data=prwa.est.evergreen1, type="l", lwd=3, col="darkgray")  
lines(upper~ Evergreen1km, data=prwa.est.evergreen1, type="l", lwd=3, col="darkgray")

 ^ what’s wrong with this? same as nobo problem

#landscape1km - highdev  
landscape1.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen1km + Grass1km + HighDev1km  
 + Schrubs1km + YearCat  
 , prwa.abund, mixture="NB", K=60)  
  
ND.prwa2 <-data.frame(HighDev1km=seq(min(sc$HighDev1km),max(sc$HighDev1km),length=100),Grass1km=0,Evergreen1km=0,Schrubs1km=0,YearCat=0)  
prwa.est.hdev1 <- predict(landscape1.prwa, type="state",  
newdata=ND.prwa2,appendData=TRUE)  
  
plot(Predicted~ HighDev1km, data=prwa.est.hdev1, ylim=c(0,10), type="l", lwd=3,  
xlab="Med/High developed cover within 1km of site", ylab="Est. PRWA Abundance")  
##95% confidence intervals  
lines(lower~ HighDev1km, data=prwa.est.hdev1, type="l", lwd=3, col="darkgray")  
lines(upper~ HighDev1km, data=prwa.est.hdev1, type="l", lwd=3, col="darkgray")



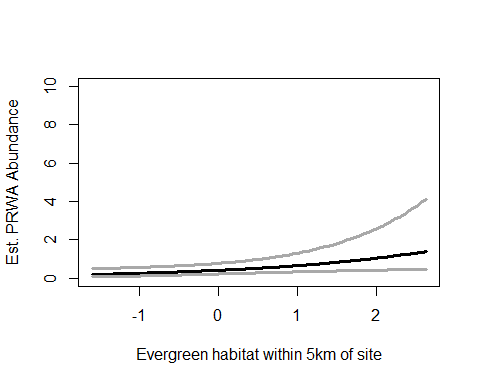
landscape5.prwa #+evergreen, - grass,

##   
## Call:  
## pcount(formula = ~Jdate + I(Jdate^2) + Noise + Time ~ Evergreen5km +   
## Grass5km + HighDev5km + Schrubs5km + YearCat, data = prwa.abund,   
## K = 60, mixture = "NB")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.863 0.315 -2.737 0.00620  
## Evergreen5km 0.456 0.188 2.428 0.01519  
## Grass5km -0.517 0.199 -2.592 0.00953  
## HighDev5km -0.299 0.272 -1.101 0.27107  
## Schrubs5km -0.142 0.149 -0.954 0.34019  
## YearCatB 0.437 0.296 1.473 0.14075  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.5513 0.366 -1.507 0.13191  
## Jdate 0.0808 0.159 0.507 0.61231  
## I(Jdate^2) -0.3669 0.175 -2.092 0.03645  
## Noise -0.2287 0.188 -1.214 0.22463  
## Time 0.4861 0.175 2.778 0.00547  
##   
## Dispersion:  
## Estimate SE z P(>|z|)  
## 8.12 26.3 0.308 0.758  
##   
## AIC: 328.7945

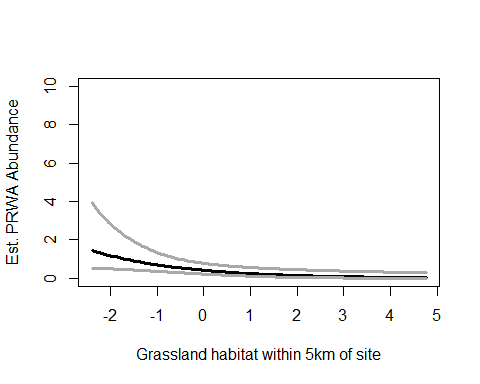
confint(landscape5.prwa, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -1.48119439 -0.2450229  
## lam(Evergreen5km) 0.08782368 0.8238262  
## lam(Grass5km) -0.90731856 -0.1260336  
## lam(HighDev5km) -0.83125459 0.2334080  
## lam(Schrubs5km) -0.43408854 0.1498979  
## lam(YearCatB) -0.14436434 1.0177667

#landscape5km - evergreen  
landscape5.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen5km + Grass5km + HighDev5km  
 + Schrubs5km + YearCat  
 , prwa.abund, mixture="NB", K=60)  
  
ND.prwa3 <-data.frame(Evergreen5km=seq(min(sc$Evergreen5km),max(sc$Evergreen5km),length=100),Grass5km=0,HighDev5km=0,Schrubs5km=0,YearCat=0)  
prwa.est.evergreen5 <- predict(landscape5.prwa, type="state",  
newdata=ND.prwa3,appendData=TRUE)  
  
plot(Predicted~ Evergreen5km, data=prwa.est.evergreen5, ylim=c(0,10), type="l", lwd=3,  
xlab="Evergreen habitat within 5km of site", ylab="Est. PRWA Abundance")  
##95% confidence intervals  
lines(lower~ Evergreen5km, data=prwa.est.evergreen5, type="l", lwd=3, col="darkgray")  
lines(upper~ Evergreen5km, data=prwa.est.evergreen5, type="l", lwd=3, col="darkgray")

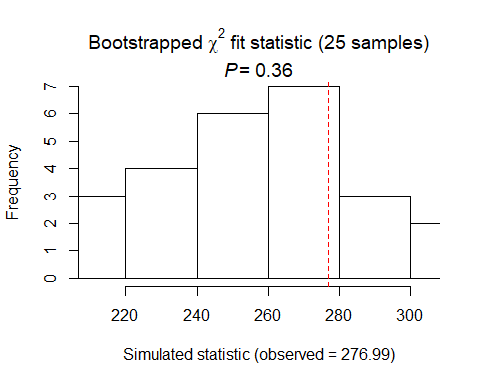


#landscape5km - grass  
landscape5.prwa <- pcount(~ Jdate + I(Jdate^2) + Noise + Time  
 ~ Evergreen5km + Grass5km + HighDev5km  
 + Schrubs5km + YearCat  
 , prwa.abund, mixture="NB", K=60)  
  
ND.prwa4 <-data.frame(Grass5km=seq(min(sc$Grass5km),max(sc$Grass5km),length=100),Evergreen5km=0,HighDev5km=0,Schrubs5km=0,YearCat=0)  
prwa.est.grass5 <- predict(landscape5.prwa, type="state",  
newdata=ND.prwa4,appendData=TRUE)  
  
plot(Predicted~ Grass5km, data=prwa.est.grass5, ylim=c(0,10), type="l", lwd=3,  
xlab="Grassland habitat within 5km of site", ylab="Est. PRWA Abundance")  
##95% confidence intervals  
lines(lower~ Grass5km, data=prwa.est.grass5, type="l", lwd=3, col="darkgray")  
lines(upper~ Grass5km, data=prwa.est.grass5, type="l", lwd=3, col="darkgray")



write.table(ms.prwa@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/PRWA\_top\_models\_ms.xls",sep="\t")

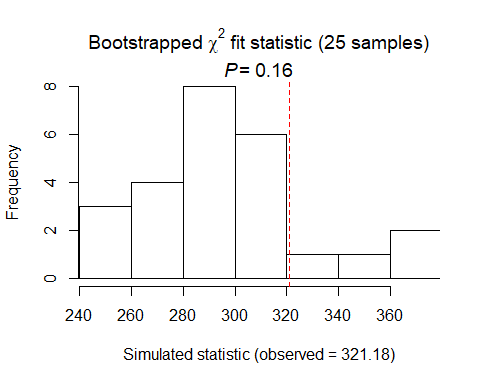
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 276.9889   
## Number of bootstrap samples = 25  
## P-value = 0.36  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 211 238 260 279 304   
##   
## Estimate of c-hat = 1.08

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 276.9889

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 321.1819   
## Number of bootstrap samples = 25  
## P-value = 0.16  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 245 274 294 313 374   
##   
## Estimate of c-hat = 1.09

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 321.1819

PRWA summary: NB distribution DCs: detect2 (+time), global2 (+time,-date2), timing(+time), time(+time) SCs using time: landscape1 (+ evergreen, - highdev) landcsape5 (+ evergreen, - grass)

# RBWO

# (forest-dwelling, insectivore, cavity-nester, bark-forager)

# covariates: tree age, treeht, canopy cover, big trees, understory doesn’t matter

rbwo.abund<- csvToUMF("rbwo\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(rbwo.abund) #det at 45 sites!  
#str(rbwo.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(rbwo.abund)= scale (obsCovs(rbwo.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(rbwo.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(rbwo.abund) <- sc

#run this when have CSV with both years  
null.rbwo <- pcount(~Jdate + Noise + Time ~1, rbwo.abund, mixture="P", K=40)  
year.rbwo <- pcount(~Jdate + Noise + Time ~ YearCat, rbwo.abund, mixture="P", K=40)  
fms.year.rbwo<- fitList(null.rbwo, year.rbwo)

## Warning in fitList(null.rbwo, year.rbwo): Your list was unnamed, so model  
## names were added as object names

year.ms.rbwo<-modSel(fms.year.rbwo)  
year.ms.rbwo

## nPars AIC delta AICwt cumltvWt  
## null.rbwo 5 642.49 0.00 0.71 0.71  
## year.rbwo 6 644.25 1.76 0.29 1.00

^ null model ranked higher but year is d1.76

#time but not date  
det.date.rbwo <- pcount(~ Jdate ~1, rbwo.abund, mixture="P", K=15)  
det.date2.rbwo <- pcount(~ Jdate + I(Jdate^2) ~1, rbwo.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.rbwo, det.date2.rbwo)

## Warning in fitList(det.date.rbwo, det.date2.rbwo): Your list was unnamed,  
## so model names were added as object names

DATEtest.rbwo <- modSel(mstestDATE)  
DATEtest.rbwo

## nPars AIC delta AICwt cumltvWt  
## det.date.rbwo 3 647.12 0.00 0.58 0.58  
## det.date2.rbwo 4 647.77 0.65 0.42 1.00

det.time.rbwo <-pcount(~ Time ~1, rbwo.abund, mixture="P",K=15)  
det.time2.rbwo <-pcount(~ Time + I(Time^2) ~1, rbwo.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.rbwo, det.time2.rbwo)

## Warning in fitList(det.time.rbwo, det.time2.rbwo): Your list was unnamed,  
## so model names were added as object names

TIMEtest.rbwo <- modSel(mstestTIME)  
TIMEtest.rbwo

## nPars AIC delta AICwt cumltvWt  
## det.time2.rbwo 4 630.81 0.00 0.987 0.99  
## det.time.rbwo 3 639.43 8.62 0.013 1.00

#detection covariates first  
det.null.rbwo <- pcount(~1 ~1, rbwo.abund, mixture="P", K=15)  
det.weather.rbwo <- pcount(~ Wind + Sky ~1, rbwo.abund, mixture="P", K=15)  
det.global2.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2) ~1, rbwo.abund, mixture="P", K=15)  
det.sound.rbwo <- pcount(~ Noise + Wind ~1, rbwo.abund, mixture="P", K=15)  
det.date.rbwo <- pcount(~ Jdate ~1, rbwo.abund, mixture="P", K=15)  
det.detect2.rbwo <- pcount(~ Jdate + Noise + Time + I(Time^2) ~1, rbwo.abund, mixture="P", K=15)  
det.notdate.rbwo <-pcount(~ Wind + Sky + Noise ~1, rbwo.abund, mixture="P", K=15)  
det.time2.rbwo <-pcount(~ Time + I(Time^2) ~1, rbwo.abund, mixture="P",K=15)  
det.timing2.rbwo <-pcount(~ Time + I(Time^2) + Jdate ~1, rbwo.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.rbwo, det.weather.rbwo, det.global2.rbwo,  
 det.sound.rbwo, det.date.rbwo, det.detect2.rbwo, det.notdate.rbwo,  
 det.time2.rbwo, det.timing2.rbwo)

## Warning in fitList(det.null.rbwo, det.weather.rbwo, det.global2.rbwo,  
## det.sound.rbwo, : Your list was unnamed, so model names were added as  
## object names

msDC.rbwo <- modSel(fmsDC)  
msDC.rbwo

## nPars AIC delta AICwt cumltvWt  
## det.global2.rbwo 8 629.91 0.00 4.5e-01 0.45  
## det.time2.rbwo 4 630.81 0.90 2.9e-01 0.74  
## det.timing2.rbwo 5 632.27 2.36 1.4e-01 0.88  
## det.detect2.rbwo 6 632.67 2.76 1.1e-01 1.00  
## det.weather.rbwo 4 642.47 12.57 8.5e-04 1.00  
## det.notdate.rbwo 5 643.43 13.53 5.3e-04 1.00  
## det.null.rbwo 2 646.08 16.17 1.4e-04 1.00  
## det.date.rbwo 3 647.12 17.21 8.3e-05 1.00  
## det.sound.rbwo 4 647.48 17.58 6.9e-05 1.00

#msDC.rbwo@Full  
#summary: global best, time one shortly after (didnt change with quadratic)

det.global2.rbwo #- with Sky, + with time-time2

##   
## Call:  
## pcount(formula = ~Jdate + Wind + Sky + Noise + Time + I(Time^2) ~   
## 1, data = rbwo.abund, K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.64 0.533 3.08 0.00209  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.7963 0.6249 -2.87 0.00404  
## Jdate -0.1368 0.0891 -1.53 0.12498  
## Wind -0.1295 0.0898 -1.44 0.14923  
## Sky -0.1745 0.0853 -2.05 0.04070  
## Noise -0.0976 0.0880 -1.11 0.26771  
## Time 0.3314 0.1033 3.21 0.00133  
## I(Time^2) -0.2782 0.0914 -3.04 0.00234  
##   
## AIC: 629.905

confint(det.global2.rbwo, type="det",method="normal")

## 0.025 0.975  
## p(Int) -3.0210601 -0.571608695  
## p(Jdate) -0.3114981 0.037957111  
## p(Wind) -0.3055468 0.046492482  
## p(Sky) -0.3416228 -0.007388506  
## p(Noise) -0.2701520 0.074980006  
## p(Time) 0.1289938 0.533868333  
## p(I(Time^2)) -0.4573244 -0.099043503

confint(det.global2.rbwo, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.5958483 2.686815

det.time2.rbwo #+ with time-time2

##   
## Call:  
## pcount(formula = ~Time + I(Time^2) ~ 1, data = rbwo.abund, K = 15,   
## mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 1.69 0.507 3.34 0.000849  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -1.828 0.5890 -3.10 0.001907  
## Time 0.367 0.0975 3.77 0.000166  
## I(Time^2) -0.279 0.0895 -3.11 0.001840  
##   
## AIC: 630.8072

confint(det.time2.rbwo, type="det",method="normal")

## 0.025 0.975  
## p(Int) -2.9826816 -0.6740311  
## p(Time) 0.1761535 0.5585338  
## p(I(Time^2)) -0.4544468 -0.1034282

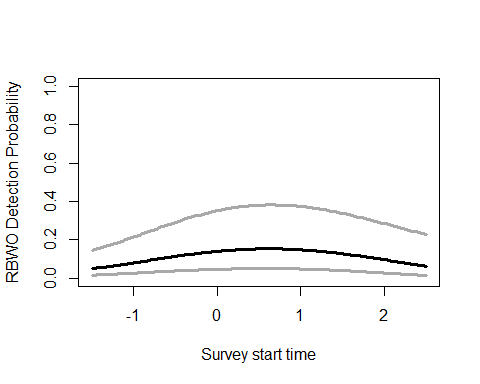
confint(det.time2.rbwo, type="state",method="normal")

## 0.025 0.975  
## lam(Int) 0.6972271 2.683166

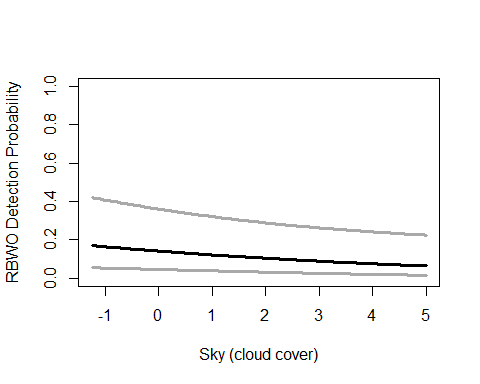
# time variable - averaged with global2, time2 model  
det.global2.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2) ~1, rbwo.abund, mixture="P", K=15)  
det.time2.rbwo <-pcount(~ Time + I(Time^2) ~1, rbwo.abund, mixture="P",K=15)  
  
#summary(obsCovs(rbwo.abund))  
dms\_top.rbwod <- fitList(det.global2.rbwo,det.time2.rbwo)

## Warning in fitList(det.global2.rbwo, det.time2.rbwo): Your list was  
## unnamed, so model names were added as object names

ND.rbwod <-data.frame(Time=seq(-1.5,2.5,length=100),Jdate=0, Wind=0,Sky=0,Noise=0)  
rbwo.est.time <- predict(dms\_top.rbwod, type="det",  
 newdata=ND.rbwod,appendData=TRUE)  
  
plot(Predicted~ Time, data=rbwo.est.time, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey start time", ylab="RBWO Detection Probability")  
##95% confidence intervals  
lines(lower~ Time, data=rbwo.est.time, type="l", lwd=3, col="darkgray")  
lines(upper~ Time, data=rbwo.est.time, type="l", lwd=3, col="darkgray")



# sky variable - only global  
det.global2.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2) ~1, rbwo.abund, mixture="P", K=15)  
  
#summary(obsCovs(rbwo.abund))  
  
ND.rbwod2 <-data.frame(Sky=seq(-1.25,5,length=100),Jdate=0, Wind=0,Time=0,Noise=0)  
rbwo.est.sky <- predict(det.global2.rbwo, type="det",  
 newdata=ND.rbwod2,appendData=TRUE)  
  
plot(Predicted~ Sky, data=rbwo.est.sky, ylim=c(0,1), type="l", lwd=3,  
 xlab="Sky (cloud cover)", ylab="RBWO Detection Probability")  
##95% confidence intervals  
lines(lower~ Sky, data=rbwo.est.sky, type="l", lwd=3, col="darkgray")  
lines(upper~ Sky, data=rbwo.est.sky, type="l", lwd=3, col="darkgray")



write.table(msDC.rbwo@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/rbwo\_top\_models\_msDC.xls",sep="\t")

#only FG\_shrub better quadratic  
testR.rbwo <- pcount(~1 ~BA, rbwo.abund, mixture="P", K=4)  
testQ.rbwo <- pcount(~1 ~BA + I(BA^2), rbwo.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.rbwo, testQ.rbwo)

## Warning in fitList(testR.rbwo, testQ.rbwo): Your list was unnamed, so model  
## names were added as object names

BAtest.rbwo <- modSel(msBAtest)  
BAtest.rbwo

## nPars AIC delta AICwt cumltvWt  
## testR.rbwo 3 665.32 0.00 0.73 0.73  
## testQ.rbwo 4 667.26 1.94 0.27 1.00

testR.rbwo <- pcount(~1 ~HW\_dens\_1050, rbwo.abund, mixture="P", K=4)  
testQ.rbwo <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), rbwo.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.rbwo, testQ.rbwo)

## Warning in fitList(testR.rbwo, testQ.rbwo): Your list was unnamed, so model  
## names were added as object names

HW1050test.rbwo <- modSel(msHW1050test)  
HW1050test.rbwo

## nPars AIC delta AICwt cumltvWt  
## testR.rbwo 3 666.50 0.00 0.53 0.53  
## testQ.rbwo 4 666.77 0.27 0.47 1.00

testR.rbwo <- pcount(~1 ~FG\_herb, rbwo.abund, mixture="P", K=4)  
testQ.rbwo <- pcount(~1 ~FG\_herb + I(FG\_herb^2), rbwo.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.rbwo, testQ.rbwo)

## Warning in fitList(testR.rbwo, testQ.rbwo): Your list was unnamed, so model  
## names were added as object names

FGHtest.rbwo <- modSel(msFGHtest)  
FGHtest.rbwo

## nPars AIC delta AICwt cumltvWt  
## testR.rbwo 3 666.39 0.00 0.73 0.73  
## testQ.rbwo 4 668.38 1.99 0.27 1.00

#yes  
testR.rbwo <- pcount(~1 ~FG\_shrub, rbwo.abund, mixture="P", K=4)  
testQ.rbwo <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), rbwo.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.rbwo, testQ.rbwo)

## Warning in fitList(testR.rbwo, testQ.rbwo): Your list was unnamed, so model  
## names were added as object names

FGStest.rbwo <- modSel(msFGStest)  
FGStest.rbwo

## nPars AIC delta AICwt cumltvWt  
## testQ.rbwo 4 665.78 0.00 0.57 0.57  
## testR.rbwo 3 666.30 0.53 0.43 1.00

testR.rbwo <- pcount(~1 ~NHW\_saplings, rbwo.abund, mixture="P", K=4)  
testQ.rbwo <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), rbwo.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.rbwo, testQ.rbwo)

## Warning in fitList(testR.rbwo, testQ.rbwo): Your list was unnamed, so model  
## names were added as object names

HWStest.rbwo <- modSel(msHWStest)  
HWStest.rbwo

## nPars AIC delta AICwt cumltvWt  
## testR.rbwo 3 666.41 0.00 0.73 0.73  
## testQ.rbwo 4 668.39 1.99 0.27 1.00

#now site covs using detection covariates (global2 DC)  
null.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2) ~1  
 , rbwo.abund, mixture="P", K=120)  
global.rbwo <- pcount(~ Jdate + Wind + Sky + Noise+ Time + I(Time^2)  
 ~ Treatment + Herbicide + BA + Nsnags +Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + FG\_herb + FG\_shrub + I(FG\_shrub^2) + NHW\_saplings + NP\_over\_20cm  
 + Rel\_HW2P\_canopy + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , rbwo.abund, mixture="P", K=120) #FPSiteIndex removed  
local.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , rbwo.abund, mixture="P", K=120) #can only include BA OR CCover  
lh.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ TreeHt + Ccover + NP\_over\_20cm + Rel\_HW2P\_canopy + YearCat  
 , rbwo.abund, mixture="P", K=120)  
landmetrics.rbwo <- pcount (~ Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Parea + ShapeIndex + YearCat  
 , rbwo.abund, mixture="P", K=120)  
landscape500.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Evergreen500m + HighDev500m + Schrubs500m + Ag500m + YearCat  
 , rbwo.abund, mixture="P", K=120)  
landscape1.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Evergreen1km + HighDev1km + Schrubs1km + YearCat  
 , rbwo.abund, mixture="P", K=120)  
landscape5.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Evergreen5km + HighDev5km + Schrubs5km + YearCat  
 , rbwo.abund, mixture="P", K=120)  
landscape30.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ Evergreen30km + HighDev30km + YearCat  
 , rbwo.abund, mixture="P", K=120) #removed Protected30  
treatment.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ Treatment + Nthins + YearCat  
 , rbwo.abund, mixture ="P", K=120)  
management.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ Treatment + BA + TimeSinceB + TimeSinceT + Herbicide + YearCat  
 , rbwo.abund, mixture="P", K=120)  
disturbance.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ TimeSinceB + TimeSinceT + YearCat  
 , rbwo.abund, mixture="P", K=120)  
siteprod.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)   
 ~ PISoils + NSoilTypes + YearCat  
 , rbwo.abund, mixture="P", K=120) #FPSiteIndex removed  
upstate.rbwo <- pcount(~ Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Parea + HighDev5km + YearCat  
 , rbwo.abund, mixture="P", K=120) #5km was pretty arbitrary  
coord.rbwo <- pcount (~Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Latitude + Longitude + YearCat  
 , rbwo.abund, mixture="P", K=80)  
  
fms <- fitList(null.rbwo, global.rbwo, local.rbwo, lh.rbwo, landmetrics.rbwo,  
 landscape500.rbwo, landscape1.rbwo, landscape5.rbwo, landscape30.rbwo,  
 treatment.rbwo, management.rbwo, disturbance.rbwo,  
 siteprod.rbwo, upstate.rbwo, coord.rbwo)

## Warning in fitList(null.rbwo, global.rbwo, local.rbwo, lh.rbwo,  
## landmetrics.rbwo, : Your list was unnamed, so model names were added as  
## object names

ms.rbwo <- modSel(fms) #remember FPSiteIndex removed from global & siteprod  
ms.rbwo

## nPars AIC delta AICwt cumltvWt  
## coord.rbwo 11 626.40 0.00 6.9e-01 0.69  
## null.rbwo 8 629.90 3.50 1.2e-01 0.81  
## local.rbwo 12 632.45 6.06 3.3e-02 0.84  
## landscape30.rbwo 11 633.19 6.79 2.3e-02 0.87  
## landscape5.rbwo 12 633.32 6.92 2.2e-02 0.89  
## landmetrics.rbwo 11 633.44 7.04 2.0e-02 0.91  
## siteprod.rbwo 11 633.60 7.20 1.9e-02 0.93  
## landscape500.rbwo 13 633.86 7.46 1.7e-02 0.94  
## treatment.rbwo 13 633.91 7.51 1.6e-02 0.96  
## landscape1.rbwo 12 634.45 8.05 1.2e-02 0.97  
## upstate.rbwo 11 634.59 8.19 1.1e-02 0.98  
## disturbance.rbwo 11 634.72 8.32 1.1e-02 0.99  
## lh.rbwo 13 636.27 9.87 5.0e-03 1.00  
## management.rbwo 16 639.27 12.87 1.1e-03 1.00  
## global.rbwo 33 659.33 32.94 4.9e-08 1.00

#ms.rbwo@Full  
#summary: coord is only top model (still)

coord.rbwo

##   
## Call:  
## pcount(formula = ~Jdate + Wind + Sky + Noise + Time + I(Time^2) ~   
## Latitude + Longitude + YearCat, data = rbwo.abund, K = 80,   
## mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) 2.985 2.6621 1.121 0.26221  
## Latitude -0.249 0.0843 -2.954 0.00314  
## Longitude -0.101 0.0813 -1.242 0.21416  
## YearCatB -0.150 0.1598 -0.936 0.34922  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -3.2352 2.7718 -1.167 0.24313  
## Jdate -0.1664 0.0850 -1.957 0.05029  
## Wind -0.1426 0.0890 -1.602 0.10908  
## Sky -0.1601 0.0806 -1.987 0.04697  
## Noise -0.0341 0.0815 -0.419 0.67511  
## Time 0.2996 0.1002 2.990 0.00279  
## I(Time^2) -0.2406 0.0855 -2.813 0.00491  
##   
## AIC: 626.3976

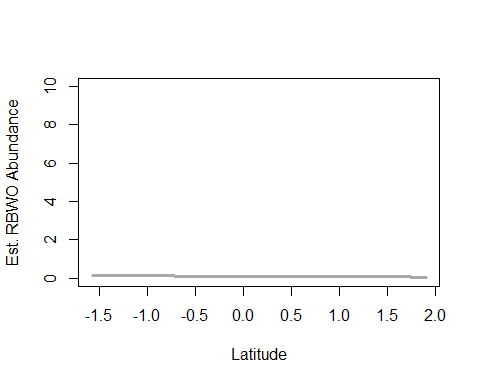
confint(coord.rbwo, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -2.2329257 8.20241163  
## lam(Latitude) -0.4140770 -0.08376536  
## lam(Longitude) -0.2604488 0.05837803  
## lam(YearCatB) -0.4626677 0.16356962

summary(sc$Latitude)

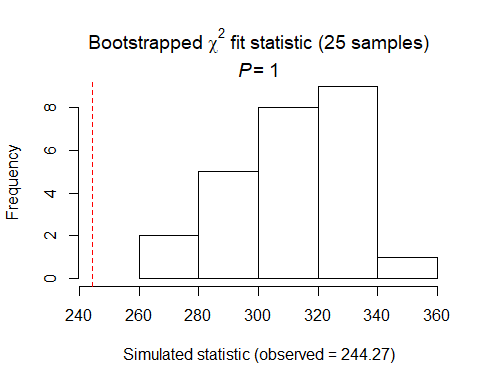
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.5810 -1.0801 0.1550 0.0000 0.8171 1.8999

#coords with latitude variable  
coord.rbwo <- pcount (~Jdate + Wind + Sky + Noise + Time + I(Time^2)  
 ~ Latitude + Longitude + YearCat  
 , rbwo.abund, mixture="P", K=80)  
  
  
ND.rbwo <-data.frame(Latitude=seq(min(sc$Latitude),max(sc$Latitude),length=100),Longitude=0,YearCat=0)  
rbwo.est.lat <- predict(coord.rbwo, type="state",  
newdata=ND.rbwo,appendData=TRUE)  
  
plot(Predicted~ Latitude, data=rbwo.est.lat, ylim=c(0,10), type="l", lwd=3,  
xlab="Latitude", ylab="Est. RBWO Abundance")  
##95% confidence intervals  
lines(lower~ Latitude, data=rbwo.est.lat, type="l", lwd=3, col="darkgray")  
lines(upper~ Latitude, data=rbwo.est.lat, type="l", lwd=3, col="darkgray")



write.table(ms.rbwo@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/rbwo\_top\_models\_ms.xls",sep="\t")

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 244.2665   
## Number of bootstrap samples = 25  
## P-value = 1  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 261 296 311 327 356   
##   
## Estimate of c-hat = 0.79

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 244.2665

RBWO Summary: P distribution DCs: global (sky -, +time&time2-, then time (+time&time2-) SCs using global2: Coord is only top model (- with Latitude)

# YBCH

# (foliage gleaner, shrub nester 1-8’ max, scrub habitat, insects)

# covariates: shrub/midstory density, forbs,

ybch.abund<- csvToUMF("ybch\_abund.csv", long = FALSE, type = "unmarkedFramePCount")  
#summary(ybch.abund)  
#str(ybch.abund)  
#scale all observation covariates (covs of detection)  
obsCovs(ybch.abund)= scale (obsCovs(ybch.abund))  
#select particular site covariates to scale below  
sc <- siteCovs(ybch.abund)  
sc[,c(6:77)] <- scale(sc[, c(6:77)])  
siteCovs(ybch.abund) <- sc

#run this when have CSV with both years  
null.ybch <- pcount(~Jdate + Noise + Time ~1, ybch.abund, mixture="P", K=40)  
year.ybch <- pcount(~Jdate + Noise + Time ~ YearCat, ybch.abund, mixture="P", K=40)  
fms.year.ybch<- fitList(null.ybch, year.ybch)

## Warning in fitList(null.ybch, year.ybch): Your list was unnamed, so model  
## names were added as object names

year.ms.ybch<-modSel(fms.year.ybch)  
year.ms.ybch

## nPars AIC delta AICwt cumltvWt  
## null.ybch 5 546.27 0.00 0.73 0.73  
## year.ybch 6 548.23 1.96 0.27 1.00

^ null ranked first but year is 1.96

#neither sig  
det.date.ybch <- pcount(~ Jdate ~1, ybch.abund, mixture="P", K=15)  
det.date2.ybch <- pcount(~ Jdate + I(Jdate^2) ~1, ybch.abund, mixture="P", K=15)  
mstestDATE <- fitList(det.date.ybch, det.date2.ybch)

## Warning in fitList(det.date.ybch, det.date2.ybch): Your list was unnamed,  
## so model names were added as object names

DATEtest.ybch <- modSel(mstestDATE)  
DATEtest.ybch

## nPars AIC delta AICwt cumltvWt  
## det.date.ybch 3 552.75 0.00 0.65 0.65  
## det.date2.ybch 4 554.01 1.26 0.35 1.00

det.time.ybch <-pcount(~ Time ~1, ybch.abund, mixture="P",K=15)  
det.time2.ybch <-pcount(~ Time + I(Time^2) ~1, ybch.abund, mixture="P",K=15)  
mstestTIME <- fitList(det.time.ybch, det.time2.ybch)

## Warning in fitList(det.time.ybch, det.time2.ybch): Your list was unnamed,  
## so model names were added as object names

TIMEtest.ybch <- modSel(mstestTIME)  
TIMEtest.ybch

## nPars AIC delta AICwt cumltvWt  
## det.time.ybch 3 565.43 0.00 0.73 0.73  
## det.time2.ybch 4 567.39 1.96 0.27 1.00

#detection covariates first  
det.null.ybch <- pcount(~1 ~1, ybch.abund, mixture="P", K=15)  
det.weather.ybch <- pcount(~ Wind + Sky ~1, ybch.abund, mixture="P", K=15)  
det.global.ybch <- pcount(~ Jdate + Wind + Sky + Noise +Time ~1, ybch.abund, mixture="P", K=15)  
det.sound.ybch <- pcount(~ Noise + Wind ~1, ybch.abund, mixture="P", K=15)  
det.date.ybch <- pcount(~ Jdate ~1, ybch.abund, mixture="P", K=15)  
det.detect.ybch <- pcount(~ Jdate + Noise + Time ~1, ybch.abund, mixture="P", K=15)  
det.notdate.ybch <-pcount(~ Wind + Sky + Noise ~1, ybch.abund, mixture="P", K=15)  
det.time.ybch <-pcount(~ Time ~1, ybch.abund, mixture="P",K=15)  
det.timing.ybch <-pcount(~ Time + Jdate ~1, ybch.abund, mixture="P", K=15)  
  
fmsDC <- fitList(det.null.ybch, det.weather.ybch, det.global.ybch,  
 det.sound.ybch, det.date.ybch, det.detect.ybch, det.notdate.ybch,  
 det.time.ybch, det.timing.ybch)

## Warning in fitList(det.null.ybch, det.weather.ybch, det.global.ybch,  
## det.sound.ybch, : Your list was unnamed, so model names were added as  
## object names

msDC.ybch <- modSel(fmsDC)  
msDC.ybch

## nPars AIC delta AICwt cumltvWt  
## det.detect.ybch 5 546.27 0.00 6.8e-01 0.68  
## det.global.ybch 7 548.12 1.85 2.7e-01 0.95  
## det.date.ybch 3 552.75 6.48 2.6e-02 0.97  
## det.timing.ybch 4 553.64 7.37 1.7e-02 0.99  
## det.sound.ybch 4 555.35 9.07 7.2e-03 1.00  
## det.notdate.ybch 5 557.30 11.03 2.7e-03 1.00  
## det.weather.ybch 4 562.09 15.81 2.5e-04 1.00  
## det.time.ybch 3 565.43 19.16 4.7e-05 1.00  
## det.null.ybch 2 566.29 20.02 3.0e-05 1.00

#msDC.ybch@Full  
#summary: 1st detect (Jdate + Noise +Time), 2nd is global (1.85)

det.detect.ybch #+ w date and - with noise

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time ~ 1, data = ybch.abund,   
## K = 15, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## 0.287 0.151 1.9 0.0577  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.5315 0.238 -2.229 0.02579  
## Jdate 0.4221 0.116 3.652 0.00026  
## Noise -0.4455 0.141 -3.150 0.00163  
## Time -0.0779 0.114 -0.686 0.49298  
##   
## AIC: 546.271

confint(det.detect.ybch, type="det",method="normal")

## 0.025 0.975  
## p(Int) -0.9988673 -0.0642185  
## p(Jdate) 0.1955544 0.6486050  
## p(Noise) -0.7226457 -0.1683451  
## p(Time) -0.3004380 0.1447231

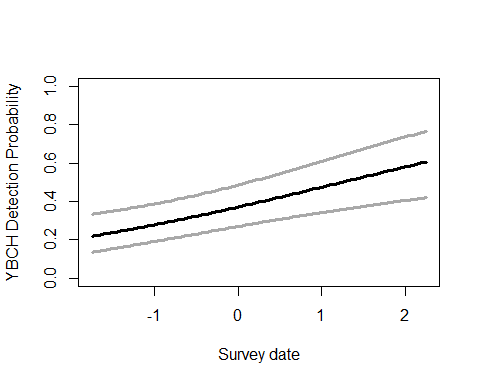
confint(det.detect.ybch, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -0.009334575 0.5826065

#+ with date - model-averaged with detect & global  
det.detect.ybch <- pcount(~ Jdate + Noise + Time ~1, ybch.abund, mixture="P", K=15)  
det.global.ybch <- pcount(~ Jdate + Wind + Sky + Noise +Time ~1, ybch.abund, mixture="P", K=15)  
  
#summary(ybch.abund)  
dms\_top.ybchd <- fitList(det.global.ybch, det.detect.ybch)

## Warning in fitList(det.global.ybch, det.detect.ybch): Your list was  
## unnamed, so model names were added as object names

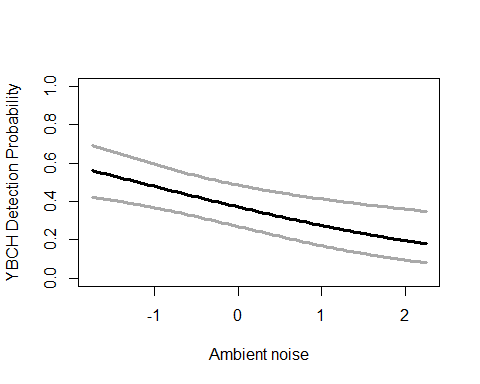
ND.ybchd <-data.frame(Jdate=seq(-1.75,2.25,length=100),Time=0, Wind=0,Sky=0,Noise=0)  
ybch.est.date <- predict(dms\_top.ybchd, type="det",  
 newdata=ND.ybchd,appendData=TRUE)  
  
plot(Predicted~ Jdate, data=ybch.est.date, ylim=c(0,1), type="l", lwd=3,  
 xlab="Survey date", ylab="YBCH Detection Probability")  
##95% confidence intervals  
lines(lower~ Jdate, data=ybch.est.date, type="l", lwd=3, col="darkgray")  
lines(upper~ Jdate, data=ybch.est.date, type="l", lwd=3, col="darkgray")



#- with noise - model-averaged with detect & global  
det.detect.ybch <- pcount(~ Jdate + Noise + Time ~1, ybch.abund, mixture="P", K=15)  
det.global.ybch <- pcount(~ Jdate + Wind + Sky + Noise +Time ~1, ybch.abund, mixture="P", K=15)  
  
#summary(obsCovs(ybch.abund))  
dms\_top.ybchd <- fitList(det.detect.ybch,det.global.ybch)

## Warning in fitList(det.detect.ybch, det.global.ybch): Your list was  
## unnamed, so model names were added as object names

ND.ybchd2 <-data.frame(Noise=seq(-1.75,2.25,length=100),Time=0,Jdate=0, Wind=0, Sky=0)  
ybch.est.noise <- predict(dms\_top.ybchd, type="det",  
 newdata=ND.ybchd2,appendData=TRUE)  
  
plot(Predicted~ Noise, data=ybch.est.noise, ylim=c(0,1), type="l", lwd=3,  
 xlab="Ambient noise", ylab="YBCH Detection Probability")  
##95% confidence intervals  
lines(lower~ Noise, data=ybch.est.noise, type="l", lwd=3, col="darkgray")  
lines(upper~ Noise, data=ybch.est.noise, type="l", lwd=3, col="darkgray")



write.table(msDC.ybch@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/ybch\_top\_models\_msDC.xls",sep="\t")

#FG herb better quadratic, only one  
testR.ybch <- pcount(~1 ~BA, ybch.abund, mixture="P", K=4)  
testQ.ybch <- pcount(~1 ~BA + I(BA^2), ybch.abund, mixture="P", K=4)  
msBAtest <- fitList(testR.ybch, testQ.ybch)

## Warning in fitList(testR.ybch, testQ.ybch): Your list was unnamed, so model  
## names were added as object names

BAtest.ybch <- modSel(msBAtest)  
BAtest.ybch

## nPars AIC delta AICwt cumltvWt  
## testR.ybch 3 554.05 0.00 0.72 0.72  
## testQ.ybch 4 555.94 1.89 0.28 1.00

testR.ybch <- pcount(~1 ~HW\_dens\_1050, ybch.abund, mixture="P", K=4)  
testQ.ybch <- pcount(~1 ~HW\_dens\_1050 + I(HW\_dens\_1050^2), ybch.abund, mixture="P", K=4)  
msHW1050test <- fitList(testR.ybch, testQ.ybch)

## Warning in fitList(testR.ybch, testQ.ybch): Your list was unnamed, so model  
## names were added as object names

HW1050test.ybch <- modSel(msHW1050test)  
HW1050test.ybch

## nPars AIC delta AICwt cumltvWt  
## testR.ybch 3 548.90 0.00 0.71 0.71  
## testQ.ybch 4 550.71 1.81 0.29 1.00

testR.ybch <- pcount(~1 ~FG\_herb, ybch.abund, mixture="P", K=4)  
testQ.ybch <- pcount(~1 ~FG\_herb + I(FG\_herb^2), ybch.abund, mixture="P", K=4)  
msFGHtest <- fitList(testR.ybch, testQ.ybch)

## Warning in fitList(testR.ybch, testQ.ybch): Your list was unnamed, so model  
## names were added as object names

FGHtest.ybch <- modSel(msFGHtest)  
FGHtest.ybch

## nPars AIC delta AICwt cumltvWt  
## testQ.ybch 4 549.89 0.00 0.71 0.71  
## testR.ybch 3 551.68 1.79 0.29 1.00

testR.ybch <- pcount(~1 ~FG\_shrub, ybch.abund, mixture="P", K=4)  
testQ.ybch <- pcount(~1 ~FG\_shrub + I(FG\_shrub^2), ybch.abund, mixture="P", K=4)  
msFGStest <- fitList(testR.ybch, testQ.ybch)

## Warning in fitList(testR.ybch, testQ.ybch): Your list was unnamed, so model  
## names were added as object names

FGStest.ybch <- modSel(msFGStest)  
FGStest.ybch

## nPars AIC delta AICwt cumltvWt  
## testR.ybch 3 567.65 0.00 0.73 0.73  
## testQ.ybch 4 569.62 1.97 0.27 1.00

testR.ybch <- pcount(~1 ~NHW\_saplings, ybch.abund, mixture="P", K=4)  
testQ.ybch <- pcount(~1 ~NHW\_saplings + I(NHW\_saplings^2), ybch.abund, mixture="P", K=4)  
msHWStest <- fitList(testR.ybch, testQ.ybch)

## Warning in fitList(testR.ybch, testQ.ybch): Your list was unnamed, so model  
## names were added as object names

HWStest.ybch <- modSel(msHWStest)  
HWStest.ybch

## nPars AIC delta AICwt cumltvWt  
## testR.ybch 3 567.46 0.00 0.52 0.52  
## testQ.ybch 4 567.59 0.13 0.48 1.00

# SCs with appropriate detection covariates (Jdate + Noise + Time )  
null.ybch <- pcount(~ Jdate + Noise + Time ~1, ybch.abund, mixture="P", K=40)  
global.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + Herbicide + BA + Ccover  
 + Ldepth + TreeHt + Age + TimeSinceB + TimeSinceT + Nthins  
 + HW\_dens\_1050 + FG\_herb + I(FG\_herb^2)  
 + FG\_shrub + NHW\_saplings  
 + Rel\_HW2P\_canopy + PISoils + NSoilTypes  
 + Parea + ShapeIndex + YearCat  
 , ybch.abund, mixture="P", K=40) #FPSiteIndex, snags,   
local.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Ccover + TreeHt + Ldepth + YearCat  
 , ybch.abund, mixture="P", K=40) #can only include BA OR CCover  
lh.ybch <- pcount(~ Jdate + Noise + Time  
 ~ BA + FG\_herb + I(FG\_herb^2) + FG\_shrub   
 + HW\_dens\_1050 + NHW\_saplings  
 + Rel\_HW2P\_canopy + YearCat  
 , ybch.abund, mixture="P", K=40)  
landmetrics.ybch <- pcount (~ Jdate + Noise + Time  
 ~ Parea + ShapeIndex + YearCat  
 , ybch.abund, mixture="P",K=40)  
landscape500.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen500m + Grass500m + HighDev500m + Schrubs500m  
 + Ag500m + OpenDev500m + YearCat  
 , ybch.abund, mixture="P", K=40)  
landscape1.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen1km + Grass1km + HighDev1km + Schrubs1km  
 + OpenDev1km + YearCat  
 , ybch.abund, mixture="P", K=40)  
landscape5.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen5km + Grass5km + HighDev5km + Schrubs5km + YearCat  
 , ybch.abund, mixture="P", K=40)  
landscape30.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen30km + Grass30km + YearCat  
 , ybch.abund, mixture="P", K=40)  
treatment.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + Nthins + YearCat  
 , ybch.abund, mixture ="P", K=40)  
management.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Treatment + BA + TimeSinceB + TimeSinceT   
 + Herbicide + YearCat  
 , ybch.abund, mixture="P", K=40)  
disturbance.ybch <- pcount(~ Jdate + Noise + Time  
 ~ TimeSinceB + TimeSinceT + YearCat  
 , ybch.abund, mixture="P", K=40)  
siteprod.ybch <- pcount(~ Jdate + Noise + Time ~ PISoils + NSoilTypes + YearCat  
 , ybch.abund, mixture="P", K=40) #FPSiteIndex  
#upstate.ybch <- pcount(~ Jdate + Noise + Time ~ X + Y + Z, ybch.abund, mixture="P", K=40)  
coord.ybch <- pcount (~Jdate + Noise + Time ~ Latitude + Longitude + YearCat  
 , ybch.abund, mixture="P", K=40)  
  
  
fmsYBCH <- fitList(null.ybch, global.ybch, local.ybch, lh.ybch, landmetrics.ybch,  
 landscape500.ybch, landscape1.ybch, landscape5.ybch, landscape30.ybch,  
 treatment.ybch, management.ybch, disturbance.ybch,  
 siteprod.ybch, coord.ybch)

## Warning in fitList(null.ybch, global.ybch, local.ybch, lh.ybch,  
## landmetrics.ybch, : Your list was unnamed, so model names were added as  
## object names

ms.ybch <- modSel(fmsYBCH) #note this does not include upstate  
ms.ybch

## nPars AIC delta AICwt cumltvWt  
## landscape1.ybch 11 515.62 0.00 9.8e-01 0.98  
## landscape5.ybch 10 524.07 8.46 1.4e-02 0.99  
## lh.ybch 13 526.04 10.43 5.3e-03 1.00  
## landscape30.ybch 8 528.33 12.72 1.7e-03 1.00  
## landscape500.ybch 12 532.62 17.00 2.0e-04 1.00  
## local.ybch 9 534.77 19.16 6.8e-05 1.00  
## management.ybch 13 538.25 22.63 1.2e-05 1.00  
## disturbance.ybch 8 541.55 25.93 2.3e-06 1.00  
## coord.ybch 8 543.51 27.89 8.6e-07 1.00  
## global.ybch 28 544.26 28.65 5.9e-07 1.00  
## null.ybch 5 546.27 30.66 2.2e-07 1.00  
## landmetrics.ybch 8 549.59 33.98 4.1e-08 1.00  
## treatment.ybch 10 549.76 34.14 3.8e-08 1.00  
## siteprod.ybch 8 552.15 36.53 1.1e-08 1.00

#ms.ybch@Full  
#Summary: Landscape1 is only top model (same still)

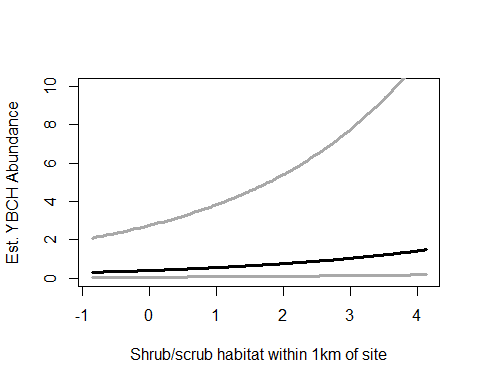
landscape1.ybch #pos with schrubs, - with opendev

##   
## Call:  
## pcount(formula = ~Jdate + Noise + Time ~ Evergreen1km + Grass1km +   
## HighDev1km + Schrubs1km + OpenDev1km + YearCat, data = ybch.abund,   
## K = 40, mixture = "P")  
##   
## Abundance:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.8901 0.9702 -0.9175 0.358906  
## Evergreen1km 0.2022 0.1370 1.4758 0.139997  
## Grass1km 0.0847 0.1161 0.7292 0.465859  
## HighDev1km -5.6553 4.3506 -1.2999 0.193638  
## Schrubs1km 0.3117 0.0863 3.6131 0.000303  
## OpenDev1km -0.3606 0.1240 -2.9084 0.003632  
## YearCatB 0.0133 0.2006 0.0664 0.947072  
##   
## Detection:  
## Estimate SE z P(>|z|)  
## (Intercept) -0.7941 0.315 -2.523 0.01162  
## Jdate 0.3327 0.112 2.982 0.00286  
## Noise -0.3184 0.131 -2.424 0.01533  
## Time -0.0999 0.107 -0.937 0.34854  
##   
## AIC: 515.616

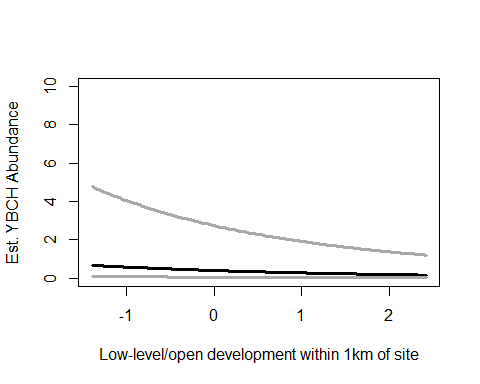
confint(landscape1.ybch, type="state",method="normal")

## 0.025 0.975  
## lam(Int) -2.79172168 1.0114654  
## lam(Evergreen1km) -0.06634202 0.4707828  
## lam(Grass1km) -0.14287696 0.3121923  
## lam(HighDev1km) -14.18234805 2.8717159  
## lam(Schrubs1km) 0.14262504 0.4808229  
## lam(OpenDev1km) -0.60366280 -0.1176076  
## lam(YearCatB) -0.37982908 0.4064606

# landscape 1: schrubs1km+  
landscape1.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen1km + Grass1km + HighDev1km + Schrubs1km  
 + OpenDev1km + YearCat  
 , ybch.abund, mixture="P", K=40)  
  
ND.ybch <-data.frame(Schrubs1km=seq(min(sc$Schrubs1km),max(sc$Schrubs1km),length=100),OpenDev1km=0,Evergreen1km=0, Grass1km=0,HighDev1km=0,YearCat=0)  
ybch.est.schrubs1 <- predict(landscape1.ybch, type="state",  
newdata=ND.ybch,appendData=TRUE)  
  
plot(Predicted~ Schrubs1km, data=ybch.est.schrubs1, ylim=c(0,10), type="l", lwd=3,  
xlab="Shrub/scrub habitat within 1km of site", ylab="Est. YBCH Abundance")  
##95% confidence intervals  
lines(lower~ Schrubs1km, data=ybch.est.schrubs1, type="l", lwd=3, col="darkgray")  
lines(upper~ Schrubs1km, data=ybch.est.schrubs1, type="l", lwd=3, col="darkgray")

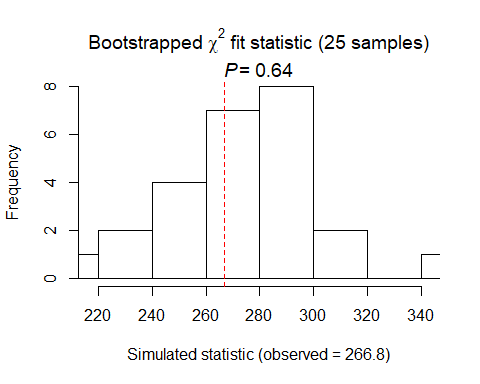


# landscape 1: opendev1km+  
landscape1.ybch <- pcount(~ Jdate + Noise + Time  
 ~ Evergreen1km + Grass1km + HighDev1km + Schrubs1km  
 + OpenDev1km + YearCat  
 , ybch.abund, mixture="P", K=40)  
  
ND.ybch2 <-data.frame(OpenDev1km=seq(min(sc$OpenDev1km),max(sc$OpenDev1km),length=100),Schrubs1km=0,Evergreen1km=0, Grass1km=0,HighDev1km=0,YearCat=0)  
ybch.est.odev1 <- predict(landscape1.ybch, type="state",  
newdata=ND.ybch2,appendData=TRUE)  
  
plot(Predicted~ OpenDev1km, data=ybch.est.odev1, ylim=c(0,10), type="l", lwd=3,  
xlab="Low-level/open development within 1km of site", ylab="Est. YBCH Abundance")  
##95% confidence intervals  
lines(lower~ OpenDev1km, data=ybch.est.odev1, type="l", lwd=3, col="darkgray")  
lines(upper~ OpenDev1km, data=ybch.est.odev1, type="l", lwd=3, col="darkgray")



write.table(ms.ybch@Full, file="C:/Users/woodj/Documents/GRAD SCHOOL - CLEMSON/Project-Specific/R work/USDA-songbirds/USDA-songbirds/ybch\_top\_models\_ms.xls",sep="\t")

## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced  
  
## Warning in rbinom(M \* J, size = rep(N, each = J), prob = pvec): NAs  
## produced



##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 266.7951   
## Number of bootstrap samples = 25  
## P-value = 0.64  
##   
## Quantiles of bootstrapped statistics:  
## 0% 25% 50% 75% 100%   
## 218 260 277 289 342   
##   
## Estimate of c-hat = 0.97

##   
## Chi-square goodness-of-fit for N-mixture model of 'unmarkedFitPCount' class  
##   
## Observed chi-square statistic = 266.7951

YBCH Summary P distribution DCs: detect first (+ date, then - noise), then global SCs using detect model: Landscape1km only top model (+Scrubs, -OpenDev)