Forth and Tay

Lindesay Scott-Hayward

18 August 2016

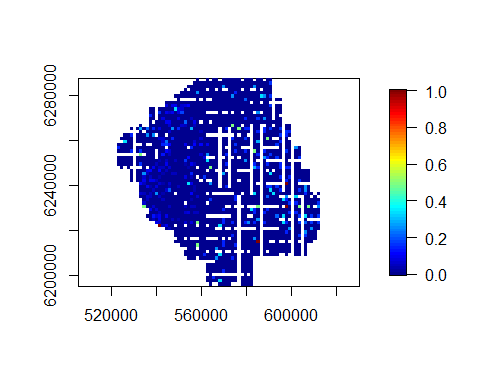
# Initial Model Fitting

### Fit Initial Model

Owing to the very low mean, the data are converted to presence/absence data and fitted using the binomial family. The panel column is created to identify each individual transect.

# convert data to presence/absence  
fat$response<-ifelse(fat$response>0, 1, 0)  
fat$panel<-as.numeric(as.factor(paste(fat$survey, fat$trip.code, sep='')))  
fat$foldid<-getCVids(fat, 5, 'panel')

require(fields)  
quilt.plot(fat$x.pos, fat$y.pos, fat$response, asp=1)



Plot of the data with mean probability of sighting per grid cell.

init\_glm<-glm(response ~ as.factor(year), data=fat, family=binomial)

### Run SALSA1D

require(mgcv)  
factorlist<-'year'  
varlist<-c('depth', 'month', 'x.pos', 'y.pos')  
  
salsa1dlist<-list(fitnessMeasure='AIC', minKnots\_1d = c(1,1,1,1), maxKnots\_1d=c(5,5,5,5), startKnots\_1d = c(1,1,1,1), degree=c(2,2,2,2), maxIterations=100, gaps=c(0,0,100,0))  
  
salsa1dout<-runSALSA1D\_withremoval(init\_glm, salsa1dlist, varlist, factorlist, varlist\_cyclicSplines = c('month'), splineParams = NULL, datain=fat, suppress.printout =TRUE)

## [1] "year will be fitted as a factor variable; there are non-zero counts for all levels"

salsa1dout$bestModel$splineParams<-salsa1dout$splineParams

summary(salsa1dout$bestModel)

##   
## Call:  
## glm(formula = response ~ as.factor(year) + bs(depth, knots = splineParams[[2]]$knots,   
## degree = splineParams[[2]]$degree, Boundary.knots = splineParams[[2]]$bd) +   
## smooth.construct(s(month, bs = "cc", k = (length(splineParams[[3]]$knots)) +   
## 2), knots = list(month = as.numeric(c(splineParams[[3]]$bd[1],   
## splineParams[[3]]$knots, splineParams[[3]]$bd[2]))),   
## data = data.frame(month))$X[, -1] + bs(x.pos, knots = splineParams[[4]]$knots,   
## degree = splineParams[[4]]$degree, Boundary.knots = splineParams[[4]]$bd) +   
## bs(y.pos, knots = splineParams[[5]]$knots, degree = splineParams[[5]]$degree,   
## Boundary.knots = splineParams[[5]]$bd), family = binomial(link = logit),   
## data = fat)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5356 -0.2402 -0.1829 -0.1390 3.3722   
##   
## Coefficients:  
## Estimate Std. Error Robust S.E. z value Pr(>|z|)   
## (Intercept) -7.4964 1.8866 1.6146 -4.643 3.44e-06 \*\*\*  
## as.factor(year)2010 -0.5350 0.1855 0.1909 -2.803 0.00507 \*\*   
## as.factor(year)2011 0.3387 0.1759 0.1844 1.837 0.06615 .   
## s(depth)1 2.6028 1.7435 1.5088 1.725 0.08451 .   
## s(depth)2 1.9420 1.4485 1.2882 1.507 0.13169   
## s(depth)3 1.1001 2.9440 2.4801 0.444 0.65734   
## s(month)1 0.7016 0.1985 0.1936 3.624 0.00029 \*\*\*  
## s(month)2 0.5628 0.1702 0.1721 3.271 0.00107 \*\*   
## s(month)3 1.5487 0.1790 0.1758 8.811 < 2e-16 \*\*\*  
## s(month)4 -0.4672 0.1695 0.1806 -2.587 0.00967 \*\*   
## s(x.pos)1 1.2885 1.4730 1.3110 0.983 0.32571   
## s(x.pos)2 -0.6640 1.0388 0.8973 -0.740 0.45931   
## s(x.pos)3 0.6622 1.1681 1.0385 0.638 0.52373   
## s(x.pos)4 1.0485 1.0816 0.9513 1.102 0.27039   
## s(y.pos)1 1.1028 0.7394 0.7035 1.567 0.11701   
## s(y.pos)2 0.9566 0.3503 0.3548 2.696 0.00701 \*\*   
## s(y.pos)3 -0.8491 1.3066 1.3326 -0.637 0.52402   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3268.8 on 15140 degrees of freedom  
## Residual deviance: 3079.4 on 15124 degrees of freedom  
## AIC: 3113.4  
##   
## Max Panel Size = 1; Number of panels = 15141  
## Number of Fisher Scoring iterations: 7

anova(salsa1dout$bestModel)

## Analysis of 'Wald statistic' Table  
## Model: binomial, link: logit  
## Response: response  
## Marginal Testing  
## Max Panel Size = 1; Number of panels = 15141  
##   
## Df X2 P(>|Chi|)   
## as.factor(year) 2 25.499 2.904e-06 \*\*\*  
## s(depth) 3 4.420 0.21954   
## s(month) 4 94.623 < 2.2e-16 \*\*\*  
## s(x.pos) 4 55.435 2.633e-11 \*\*\*  
## s(y.pos) 3 9.108 0.02789 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Depth is not significant so this is removed from the model and the process repeated:

factorlist<-'year'  
varlist<-c('month', 'x.pos', 'y.pos')  
  
salsa1dlist<-list(fitnessMeasure='AIC', minKnots\_1d = c(1,1,1), maxKnots\_1d=c(5,5,5), startKnots\_1d = c(1,1,1), degree=c(2,2,2), maxIterations=100, gaps=c(0,100,0))  
  
salsa1dout<-runSALSA1D\_withremoval(init\_glm, salsa1dlist, varlist, factorlist, varlist\_cyclicSplines = c('month'), splineParams = NULL, datain=fat, suppress.printout =TRUE)

## [1] "year will be fitted as a factor variable; there are non-zero counts for all levels"

salsa1dout$bestModel$splineParams<-salsa1dout$splineParams

summary(salsa1dout$bestModel)

##   
## Call:  
## glm(formula = response ~ as.factor(year) + smooth.construct(s(month,   
## bs = "cc", k = (length(splineParams[[2]]$knots)) + 2), knots = list(month = as.numeric(c(splineParams[[2]]$bd[1],   
## splineParams[[2]]$knots, splineParams[[2]]$bd[2]))), data = data.frame(month))$X[,   
## -1] + bs(x.pos, knots = splineParams[[3]]$knots, degree = splineParams[[3]]$degree,   
## Boundary.knots = splineParams[[3]]$bd) + bs(y.pos, knots = splineParams[[4]]$knots,   
## degree = splineParams[[4]]$degree, Boundary.knots = splineParams[[4]]$bd),   
## family = binomial(link = logit), data = fat)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.4769 -0.2486 -0.1835 -0.1367 3.3682   
##   
## Coefficients:  
## Estimate Std. Error Robust S.E. z value Pr(>|z|)   
## (Intercept) -6.0467 0.8652 0.9080 -6.659 2.75e-11 \*\*\*  
## as.factor(year)2010 -0.4393 0.1845 0.1889 -2.325 0.02007 \*   
## as.factor(year)2011 0.2649 0.1839 0.1920 1.380 0.16758   
## s(month)1 1.6927 0.2313 0.2430 6.965 3.29e-12 \*\*\*  
## s(month)2 0.9326 0.1712 0.1748 5.334 9.61e-08 \*\*\*  
## s(month)3 1.7566 0.2131 0.2244 7.830 4.89e-15 \*\*\*  
## s(x.pos)1 0.8262 1.1618 1.2493 0.661 0.50840   
## s(x.pos)2 0.2946 0.7295 0.7676 0.384 0.70109   
## s(x.pos)3 -0.5819 0.7926 0.8428 -0.690 0.48989   
## s(x.pos)4 0.8333 0.8059 0.8432 0.988 0.32305   
## s(x.pos)5 1.3925 0.7948 0.8344 1.669 0.09515 .   
## s(y.pos)1 1.2711 0.6592 0.6651 1.911 0.05599 .   
## s(y.pos)2 0.9799 0.3161 0.3164 3.097 0.00195 \*\*   
## s(y.pos)3 -0.7431 1.2682 1.2710 -0.585 0.55877   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3268.8 on 15140 degrees of freedom  
## Residual deviance: 3085.6 on 15127 degrees of freedom  
## AIC: 3113.6  
##   
## Max Panel Size = 1; Number of panels = 15141  
## Number of Fisher Scoring iterations: 7

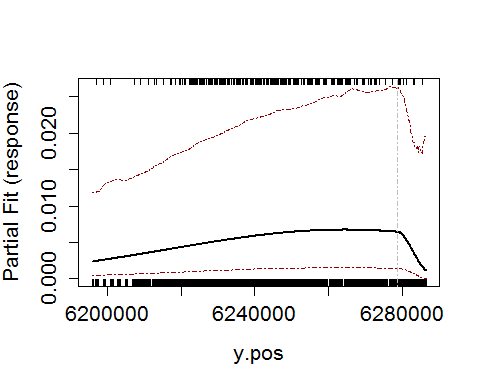
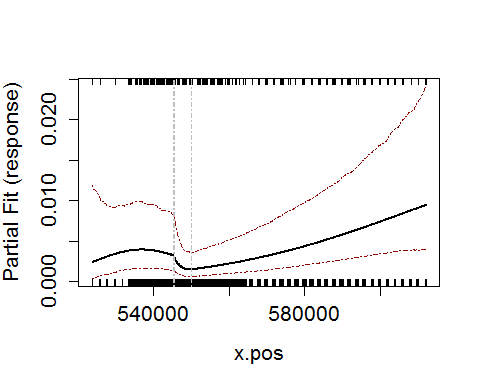
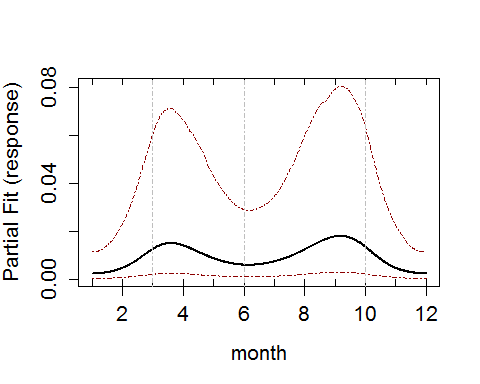
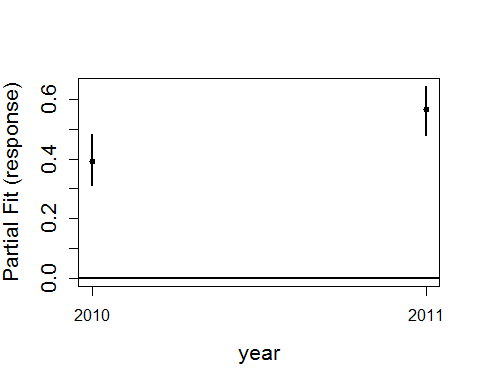
anova(salsa1dout$bestModel)

## Analysis of 'Wald statistic' Table  
## Model: binomial, link: logit  
## Response: response  
## Marginal Testing  
## Max Panel Size = 1; Number of panels = 15141  
##   
## Df X2 P(>|Chi|)   
## as.factor(year) 2 18.894 7.894e-05 \*\*\*  
## s(month) 3 65.316 4.297e-14 \*\*\*  
## s(x.pos) 5 61.196 6.877e-12 \*\*\*  
## s(y.pos) 3 11.285 0.01028 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

bestModel<-make.gamMRSea(salsa1dout$bestModel, panelid = fat$panel, splineParams = salsa1dout$splineParams, varshortnames = varlist, gamMRSea=TRUE)  
#splineParams<-bestModel$splineParams

require(mgcv)  
runPartialPlots(bestModel, data = fat, factorlist.in = factorlist, varlist.in = varlist, showKnots = T)

## [1] "Making partial plots"



### Runs Test Check

Data generated under the null hypothesis of the runs test; independence and with no change induced at this stage.

nsim<-500  
newdat<-generateNoise(nsim, fitted(bestModel), family='zibinomial', size=1)

500 sets of noisy data are simulated from the model using an overdispersed Poisson distribution where $ = $ 1.

empdistribution<-getRunsCritVals(n.sim = nsim, simData=newdat,   
 model = bestModel, data = fat, plot=TRUE,   
 returnDist = TRUE, dots=FALSE)

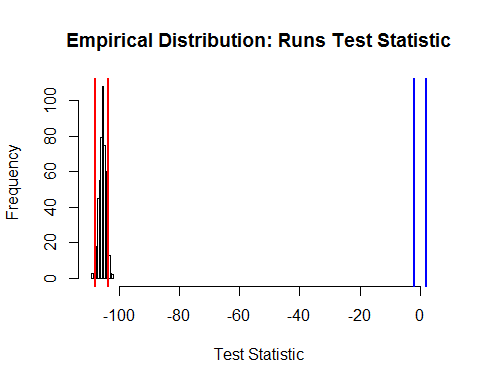


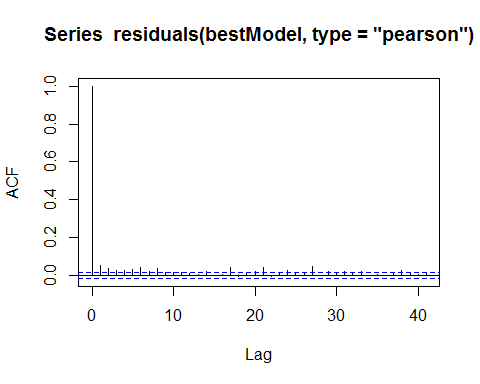
Figure showing the distribution of test statistics from a runs test. The red lines show the lower 2.5% and upper 97.5% critical values of the empirical distribution and the blue lines are from the Normal () distribution.

Evaluate the runs test using the empirical distribution to determine if the data are independent

runs.test(residuals(bestModel, type='pearson'), critvals = empdistribution)

##   
## Runs Test - Two sided; Empirical Distribution  
##   
## data: residuals(bestModel, type = "pearson")  
## Standardized Runs Statistic = -106.25, p-value = 0.196

acf(residuals(bestModel, type='pearson'))



The residuals are considered independent so for the power analysis, the data will be generated as independent as already done.

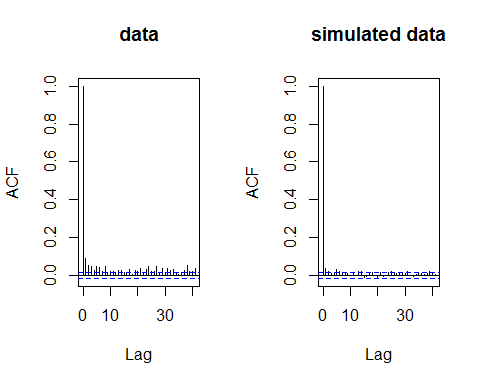
# Data Generation

Data are generated with no change but with the same correlation as the original data so that the data generation process can be checked.

Even though the data are considered independent, the following generates data with the same correlation structure and shows that the IC method does not induce correlation in the simulated data.

corrs<-getCorrelationMat(fat$panel, fat$response, dots=FALSE)  
newdatcor<-generateIC(data = fat, corrs = corrs, panels = 'panel', newdata = newdat, nsim = nsim, dots = FALSE)

par(mfrow=c(1,2))  
acf(fat$response, main='data')  
acf(newdatcor[,1], main='simulated data')

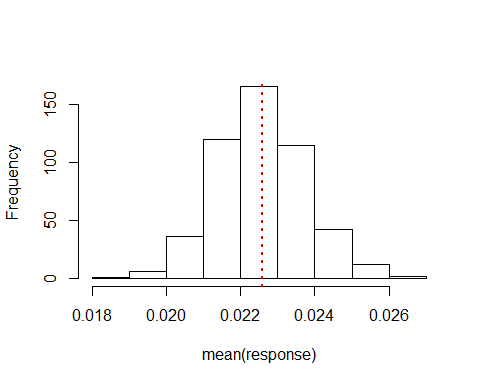


### Check simulated data consistency

The generated data are consistent with the original data:

#### Mean

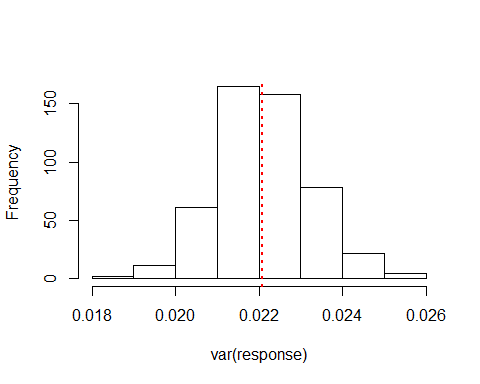
hist(apply(newdat, 2, mean), main='', xlab='mean(response)')  
abline(v=mean(fat$response), col='red', lwd=2, lty=3)



Histogram of the mean of the simulated data, with the red line representing the mean of the original data.

#### Variance

hist(apply(newdat, 2, var), main='', xlab='var(response)')  
abline(v=var(fat$response), col='red', lwd=2, lty=3)



Histogram of the variance of the simulated data, with the red line representing the variance of the original data.

var(fat$response)

## [1] 0.02207893

#### Data distribution

par(mfrow=c(2,2))  
quilt.plot(fat$x.pos, fat$y.pos, fat$response, main='Original Data', asp=1)  
quilt.plot(fat$x.pos, fat$y.pos, fitted(bestModel), main='Fitted Values', asp=1)  
quilt.plot(fat$x.pos, fat$y.pos, newdat[,1], main='Simulated Data A', asp=1)  
quilt.plot(fat$x.pos, fat$y.pos, newdat[,10], main='Simulated Data B', asp=1)

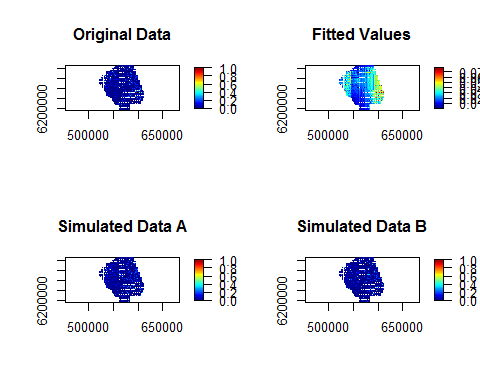


Figure showing the spatial distribution of the original data, the fitted values from the model and two examples of the simulated data.

# Power Analysis

Generate data for power analysis with impact of 50%

nsim=500  
impdata<-genOverallchangeData(-0.69, bestModel, data = fat, panels = 'panel')  
newdat<-generateNoise(nsim, impdata$truth, family='zibinomial', size=1)

Check the generated data has reduced the mean probability of sighting post impact by half:

# original data  
mean(fat$response)

## [1] 0.02258768

# fitted value mean  
mean(fitted(bestModel))

## [1] 0.02258768

# true surface  
mean(impdata$truth[impdata$eventphase==0,1])

## [1] 0.02258768

mean(impdata$truth[impdata$eventphase==1,1])

## [1] 0.01153581

# example generated surface  
mean(newdat[impdata$eventphase==0,1])

## [1] 0.02252163

mean(newdat[impdata$eventphase==1,1])

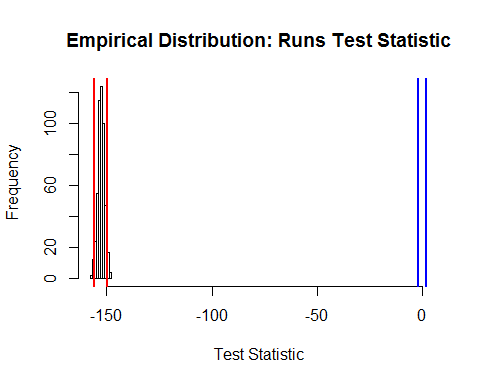
## [1] 0.01169011

Update the initial model to include the eventphase term, indicating overall change.

require(mgcv)  
fatsim\_glm<-update(bestModel, .~. + eventphase, data=impdata)

Using the simulated data with noise but no correlation, estimate the new values for the empirical distribution for the runs test.

# make sure that the independent data is used to get the null distribution  
empdistpower<-getRunsCritVals(n.sim = nsim, simData=newdat,   
 model = fatsim\_glm, data = impdata, plot=TRUE,   
 returnDist = TRUE, dots=FALSE)



grid<-expand.grid(seq(min(fat$x.pos), max(fat$x.pos), by=1000), seq(min(fat$y.pos), max(fat$y.pos), by=1000))  
names(grid)<-c('x.pos', 'y.pos')  
  
bnd<-cbind(fat$x.pos, fat$y.pos)[chull(x = fat$x.pos, fat$y.pos),]  
require(splancs)  
gridsub<-grid[inout(grid, bnd),]  
  
predictdata<-rbind(data.frame(gridsub, month=9, year=2011, eventphase=0), data.frame(gridsub, month=9, year=2011, eventphase=1))

nsim=100  
require(splines)  
powerout<-powerSimOverallChange(newdat, fatsim\_glm, empdistpower, nsim=nsim, powercoefid=length(coef(fatsim\_glm)), predictionGrid=predictdata, g2k=NULL, splineParams=bestModel$splineParams, sigdif=TRUE, n.boot=500, impact.loc=c(570000, 6240000))

# Power Outputs:

## Summary output

null.output<-pval.coverage.null(newdat.ind = newdat, model = fatsim\_glm, nsim = 500, powercoefid = length(coef(fatsim\_glm)))

summary(powerout, null.output, truebeta=-0.69)

##   
## ++++ Summary of Power Analysis Output ++++  
##   
## Number of power simulations = 100  
## Number of no change simulations = 250   
##   
## Power to select 'change' term:  
##   
## Under Change (true parameter = -0.69) = 100%  
## Under no change (true parameter = 0) = 5.6%  
##   
## Coverage for 'change' coefficient:  
##   
## Under model = 95%  
## Under no change = 93.2%  
##   
## Overall Mean Proportion Summary with 95% Confidence Intervals:  
##   
## Mean LowerCI UpperCI  
## Before 0.08669337 0.06736875 0.10668793  
## After 0.04589569 0.03452815 0.05692273

### Proportion of significant differences:

names(powerout$significant.differences)[1]<-"individual"  
plotdata<-plot.sigdiff(powerout, predictdata[predictdata$eventphase==0,c('x.pos', 'y.pos')], tailed='two', error.rate = 0.05, family=FALSE)  
plotdata

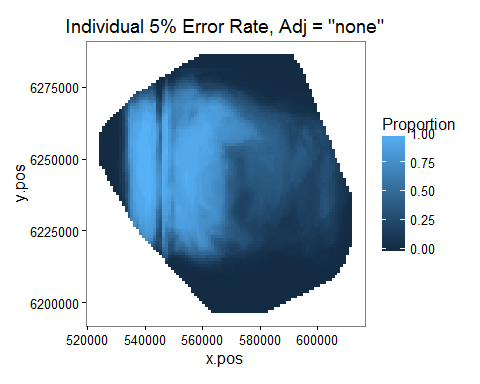


Figure showing the proportion of simulations which showed a significant difference (post event - pre event) in a given grid cell. Significance at the 5% level was determined by comparing the estimated difference in a given cell to the distribution of differences in that cell under the null hypothesis of no-change.

plotdata<-plot.sigdiff(powerout, predictdata[predictdata$eventphase==0,c('x.pos', 'y.pos')], tailed='two', error.rate = 0.05, family=TRUE)  
plotdata

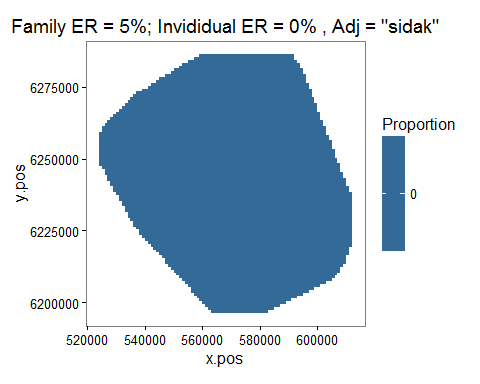


Figure showing the proportion of simulations which showed a significant difference (post event - pre event) in a given grid cell. Significance at the 5% level was determined by comparing the estimated difference in a given cell to the distribution of differences across all grid cells under the null hypothesis of no-change.

### Distance to event site plot

d2impPlot<-plot.d2imp(powerout, pct.diff=FALSE)  
d2impPlot

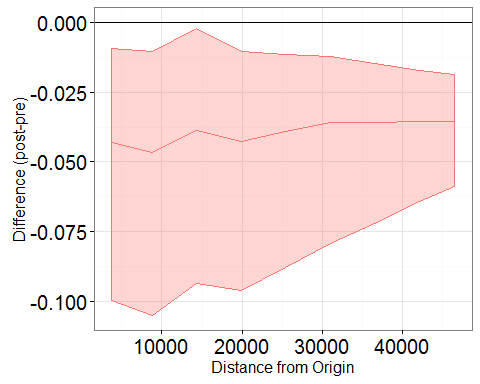


Figure showing the effect of the event on animal numbers with distance from the event site; post event - pre event. The confidence intervals are 95% bootstrap intervals.

d2impPlot<-plot.d2imp(powerout, pct.diff=T)  
d2impPlot

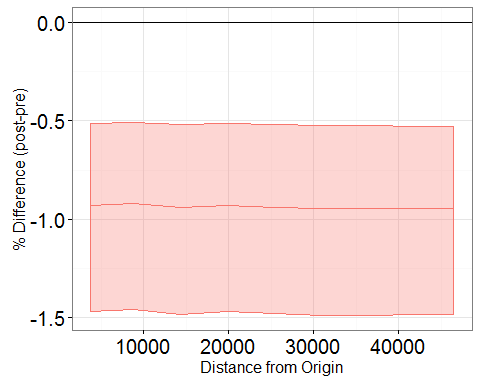


Figure showing the effect of the event on animal numbers with distance from the event site; post event - pre event. The confidence intervals are 95% bootstrap intervals.