

Lincs Power Analysis: Auks

Lindesay Scott-Hayward

2016-11-14

```
devtools::load_all(pkg='C://MarineScotlandPower/MRSeaPower')
devtools::load_all(pkg='C://MarineScotlandPower/MRSea/MRSea')
#setwd("C://MarineScotlandPower/MRSeaPower")
```

```
require(fields)
require(splines)
require(mgcv)
require(MRSea)
require(dplyr)
require(ggplot2)
require(RColorBrewer)
breaks<-ticks<-c(0, 0.2, 0.4, 0.6, 0.8, 1, 10, 25)
mypalette<-rev(brewer.pal(length (ticks)-1,"Spectral"))
require(sp)
require(raster)
require(splancs)
```

```
lincs<-read.csv('D:\\lindesay\\MarScoPower\\finalreportdata\\phase1Data.csv')
```

```
lincs$response<-lincs$AUK
lincs$foldid<-getCVids(lincs, 10, 'line_id')
```

remove data points > 300 animals

```
lincs<-lincs[-which(lincs$response>300),]
```

```
initialModel<-glm(response ~ 1 + offset(log(area)), data=lincs, family=quasipoisson)
```

```
factorlist<-NULL
varlist<-c('depth', 'month')
```

```
salsaidlist<-list(fitnessMeasure='QAIC', minKnots_1d = c(1,1), maxKnots_1d=c(5,5), startKnots_1d = c(1,
```

```
salsaidout<-runSALSA1D_withremoval(initialModel, salsaidlist, varlist, factorlist, varlist_cyclicSpline
```

```
salsaidout$bestModel$splineParams<-salsaidout$splineParams
```

```
bestModel1D<-make.gamMRSea(salsaidout$bestModel, panelid = 1:nrow(lincs), splineParams = salsaidout$spl
```

```
sink()
sink()
```

```
summary(salsaidout$bestModel)
```

```
##
```

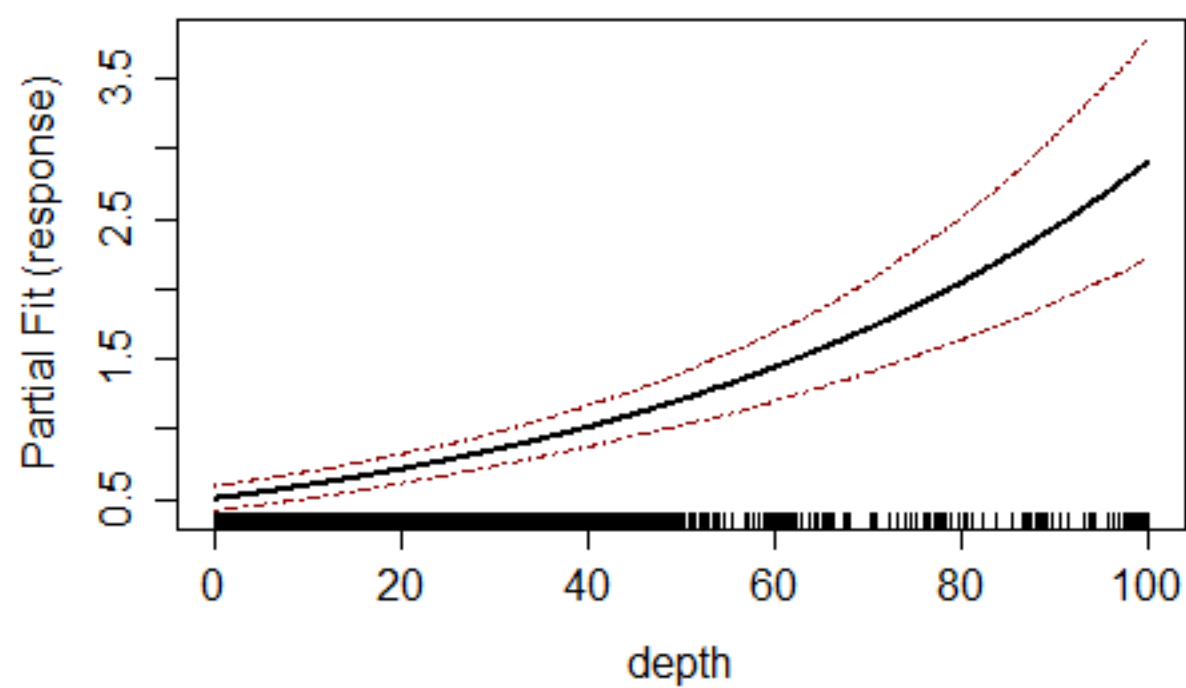
```
## Call:
## glm(formula = response ~ depth + smooth.construct(s(month, bs = "cc",
##      k = (length(splineParams[[3]]$knots) + 2), knots = list(month = as.numeric(c(splineParams[[3]]$
##      splineParams[[3]]$knots, splineParams[[3]]$bd[2]))), data = data.frame(month))$X[,
##      -1] + offset(log(area)), family = quasipoisson(link = log),
##      data = lincs)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7.985  -2.976  -2.172  -0.845   55.960
##
## Coefficients:
##              Estimate Std. Error Robust S.E. t value Pr(>|t|)
## (Intercept) -0.690053   0.093241   0.067249 -10.261 < 2e-16 ***
## depth        0.017581   0.001768   0.001382  12.717 < 2e-16 ***
## s(month)1    -0.040207   0.121832   0.079558  -0.505 0.613302
## s(month)2    -2.124289   0.286280   0.200515 -10.594 < 2e-16 ***
## s(month)3    -0.420676   0.158794   0.124797  -3.371 0.000751 ***
## s(month)4     0.597701   0.122806   0.084074   7.109 1.21e-12 ***
## s(month)5     1.347993   0.135711   0.101231  13.316 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 75.08211)
##
##      Null deviance: 290147  on 18869  degrees of freedom
## Residual deviance: 252741  on 18863  degrees of freedom
## AIC: NA
##
## Max Panel Size = 1; Number of panels = 18870
## Number of Fisher Scoring iterations: 7
```

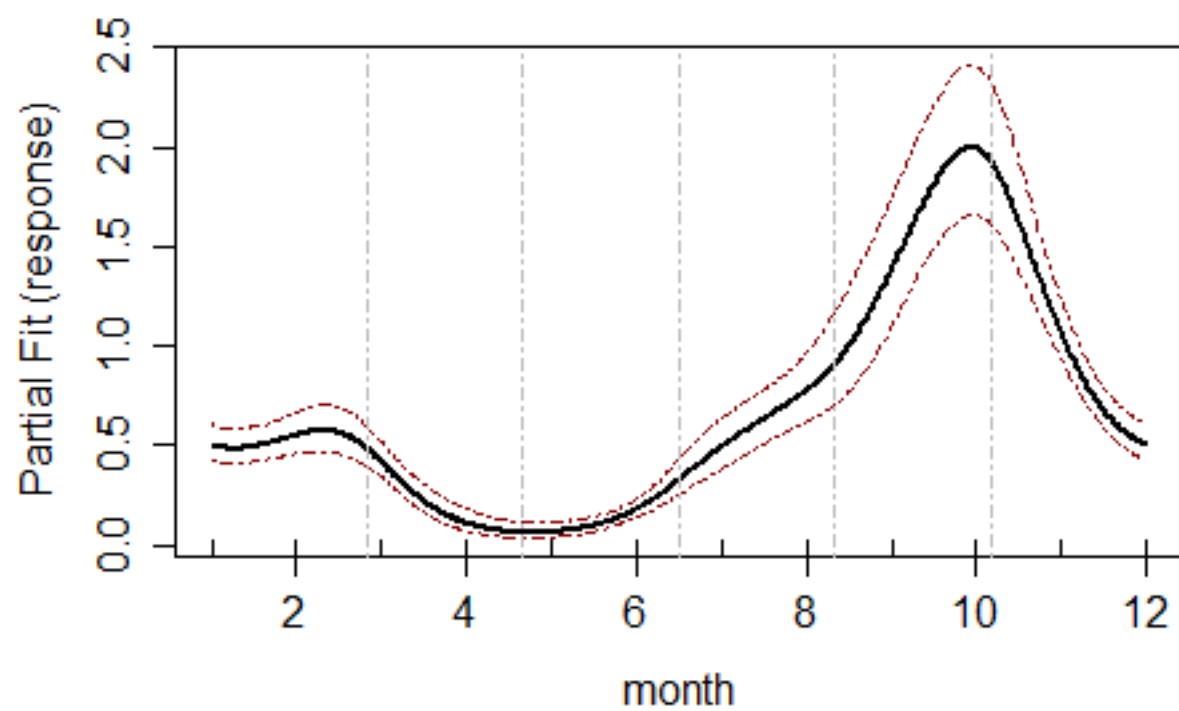
```
anova(salsaidout$bestModel)
```

```
## Analysis of 'Wald statistic' Table
## Model: quasipoisson, link: log
## Response: response
## Marginal Testing
## Max Panel Size = 1; Number of panels = 18870
##
##              Df      X2 P(>|Chi|)
## s(depth)    1 161.73 < 2.2e-16 ***
## s(month)    5 860.28 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

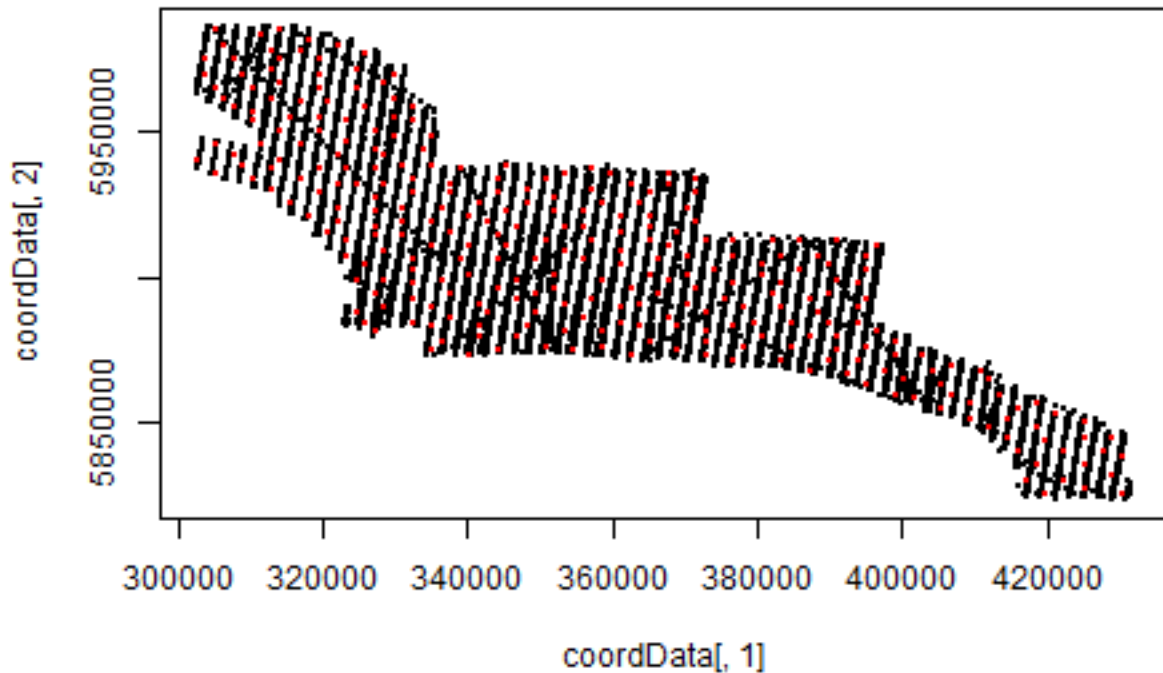
```
par(mfrow=c(1,2))
runPartialPlots(bestModel1D, data = lincs, factorlist.in = factorlist, varlist.in = varlist, showKnots =
```

```
## [1] "Making partial plots"
```





```
knotgrid<-getKnotgrid(cbind(lincs$x.pos, lincs$y.pos))
```



```
distMats<-makeDists(cbind(lincs$x.pos, lincs$y.pos), na.omit(knotgrid))
# choose sequence of radii
r_seq<-getRadiiChoices(8, distMats$dataDist)

salsa2dlist<-list(fitnessMeasure = 'QAIC', knotgrid = knotgrid, knotdim = c(100, 100), startKnots=6, m

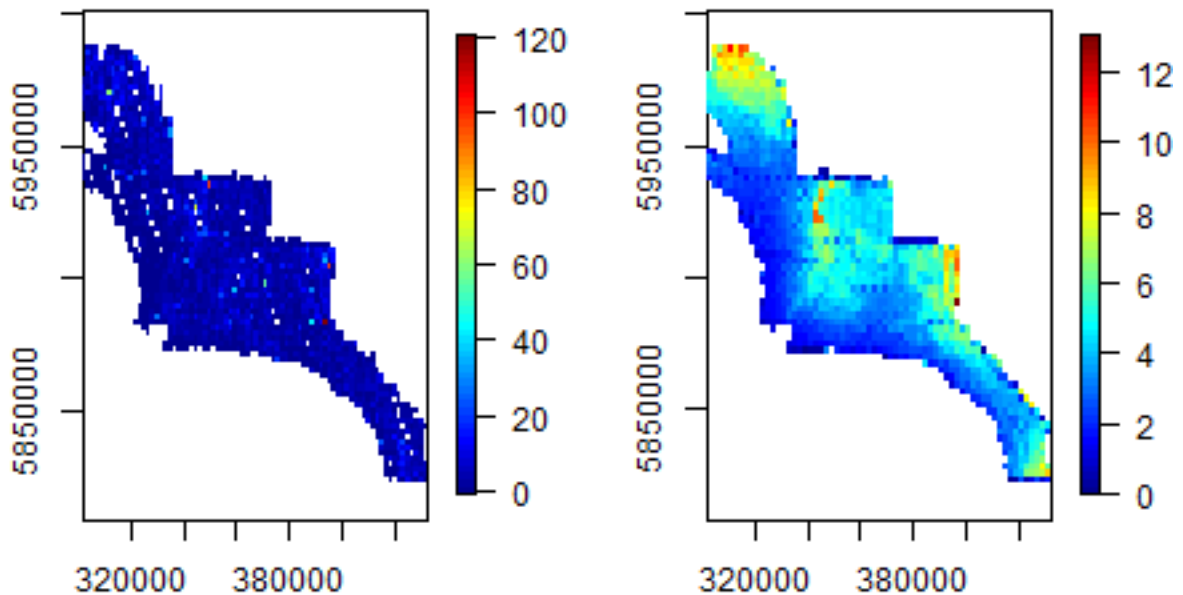
salsa2dOutput<-runSALSA2D(bestModel1D, salsa2dlist, d2k=distMats$dataDist, k2k=distMats$knotDist, splin
```

```
## [1] "initialDispersion 75.0818456312729"
## [1] "*****"
## [1] "Initialising..."
## [1] "*****"
## [1] "Space-filling knots..."
## [1] "Initialising model..."
## [1] "Initial model fitted..."
## [1] "Fitting Initial Radii"
## [1] "initialising complete"
## [1] "*****"
## [1] "Exchanging..."
## [1] "*****"
## [1] "*****"
## [1] "Moving knot..."
## [1] "*****"
## [1] "Adding knot..."
```

```
## [1] "*****"
## [1] "Improving..."
## [1] "*****"
## [1] "*****"
## [1] "Simplifying model..."
## [1] "*****"
## [1] "And we're done..."
## [1] "*****"
## [1] "Initialising..."
## [1] "*****"
## [1] "Initialising model..."
## [1] "Initial model fitted..."
## [1] "Fitting Initial Radii"
## [1] "initialising complete"
```

```
bestModel<-make.gamMRSea(salsa2dOutput$bestModel, panelid = 1:nrow(lincs), splineParams = salsa2dOutput$
```

```
par(mfrow=c(1,2))
quilt.plot(lincs$x.pos, lincs$y.pos, lincs$response, asp=1, nrow=80, ncol=80)
quilt.plot(lincs$x.pos, lincs$y.pos, fitted(bestModel), asp=1)
```



```
save(bestModel, file='D:/lindesay/MarScoPower/finalreportdata/lincsAUKmodel.RData', compress='bzip2')
```

check for correlation:

```
nsim<-500
d<-as.numeric(summary(bestModel)$dispersion)
newdat<-generateNoise(nsim, fitted(bestModel), family='poisson', d=d)

empdistribution<-getRunsCritVals(n.sim = nsim, simData=newdat, model = bestModel, data = lincs, plot=
```

##

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

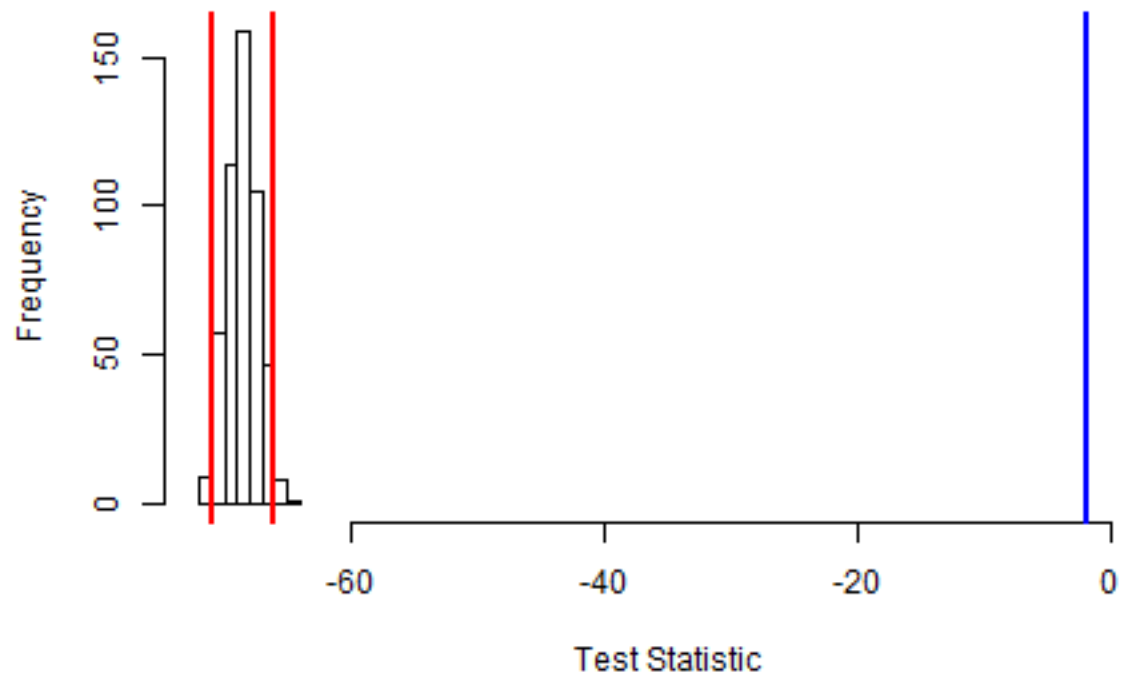
497

498

499

500

Empirical Distribution: Runs Test Statistic

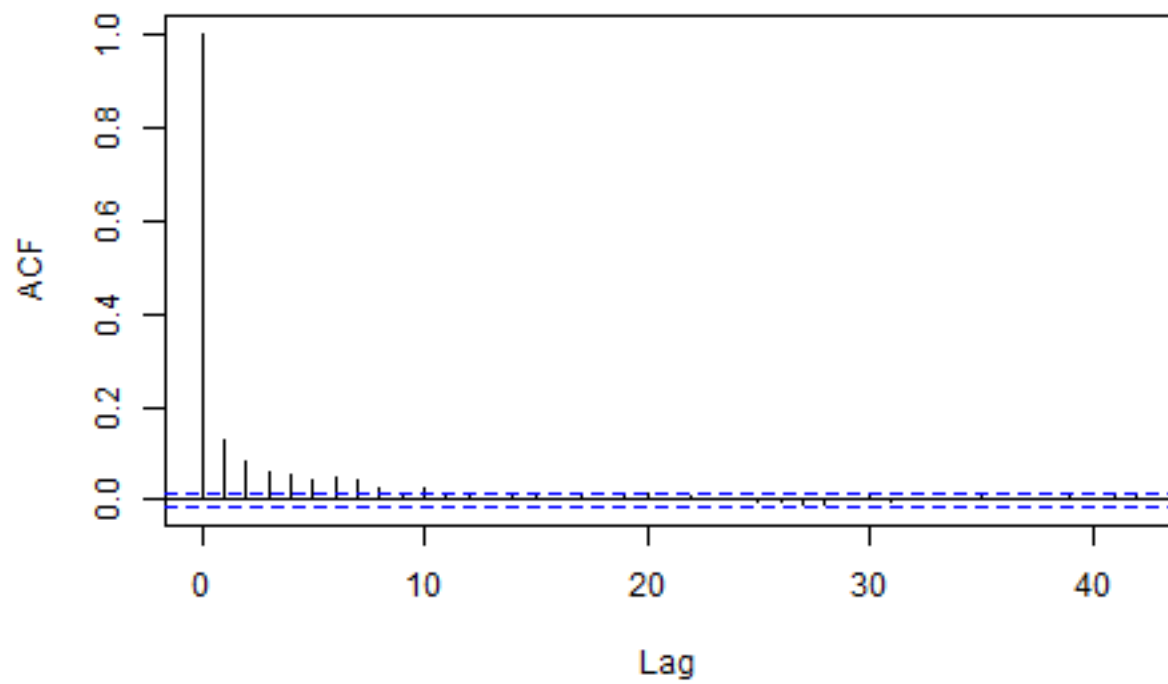


```
runs.test(residuals(bestModel, type='pearson'), critvals = empdistribution)
```

```
##  
## Runs Test - Two sided; Empirical Distribution  
##  
## data: residuals(bestModel, type = "pearson")  
## Standardized Runs Statistic = -68.271, p-value = 0.604
```

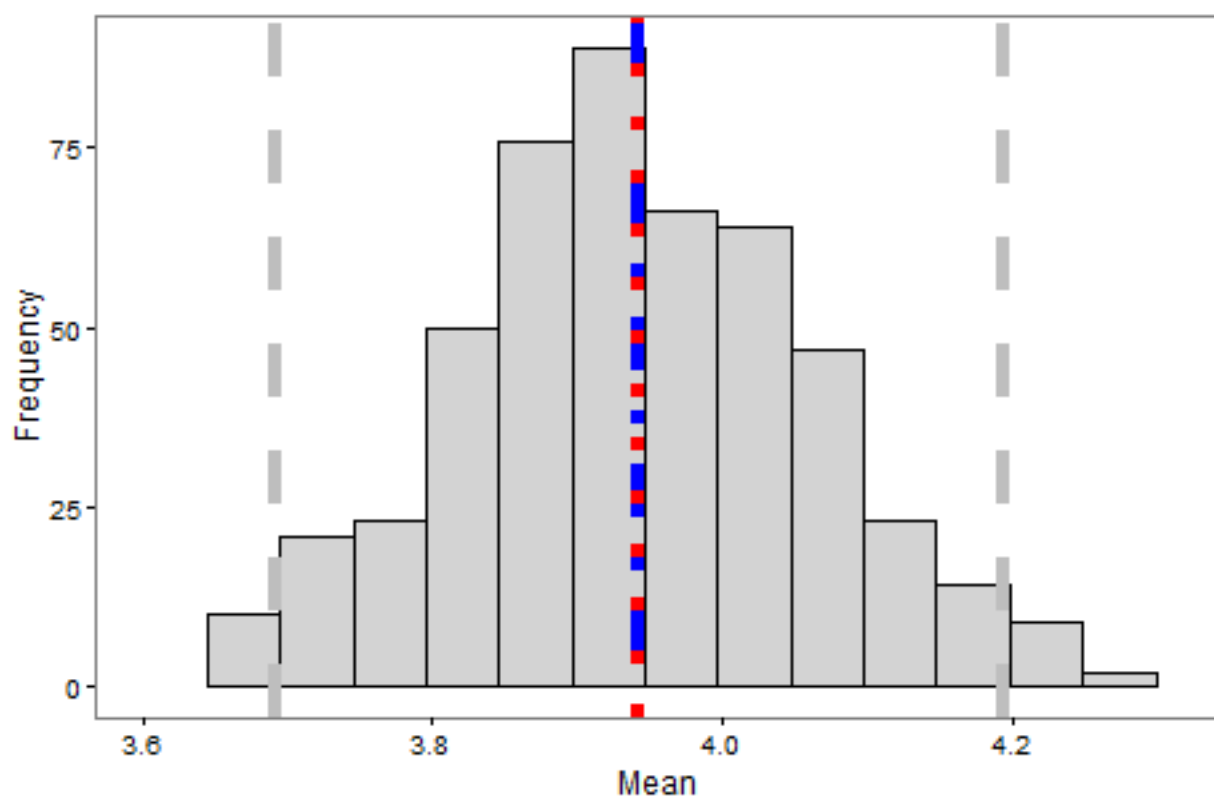
```
acf(residuals(bestModel, type='pearson'))
```

Series residuals(bestModel, type = "pearson")

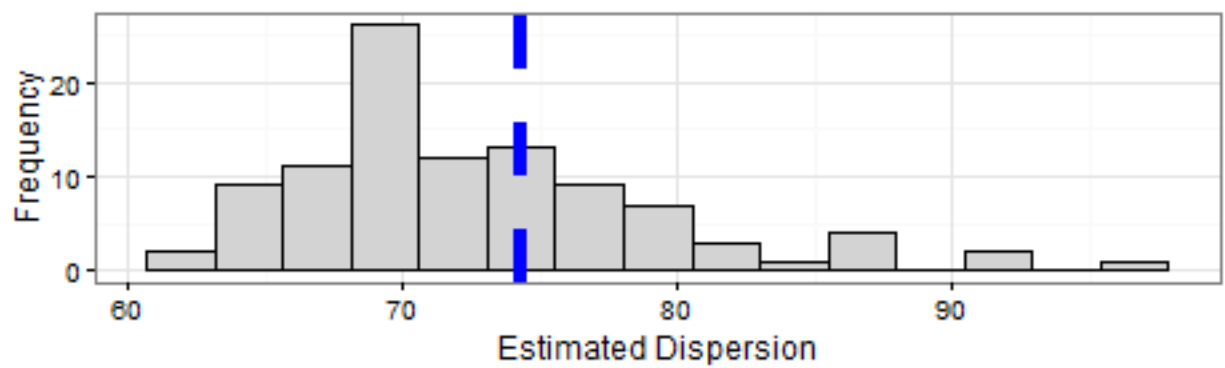
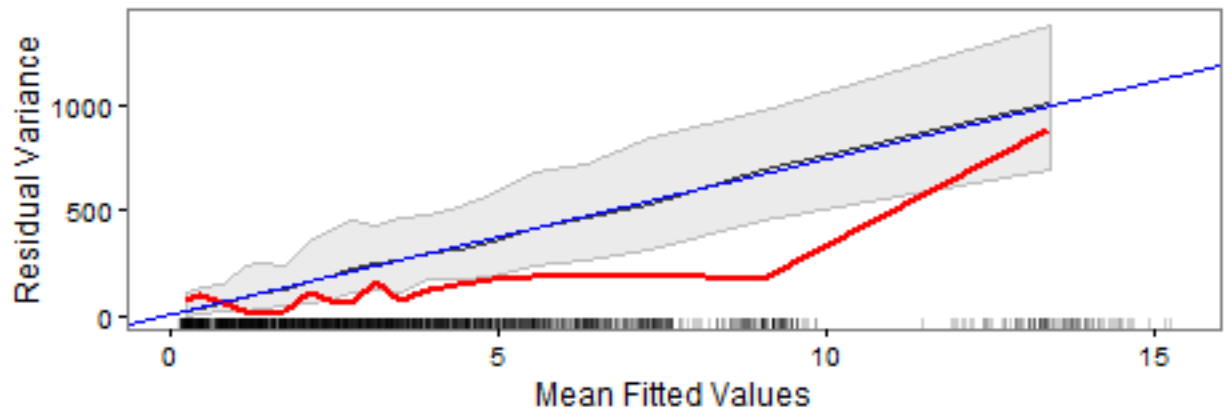


check data generation process

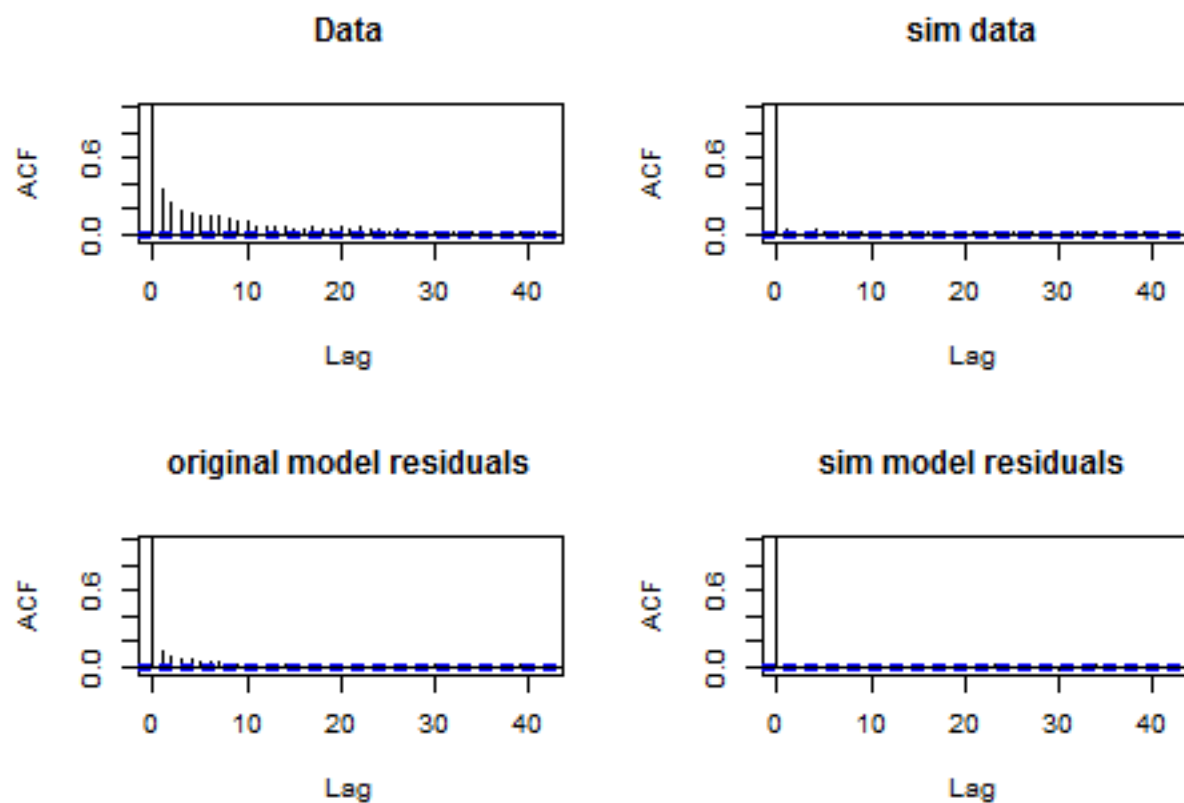
```
plotMean(bestModel, newdat)
```



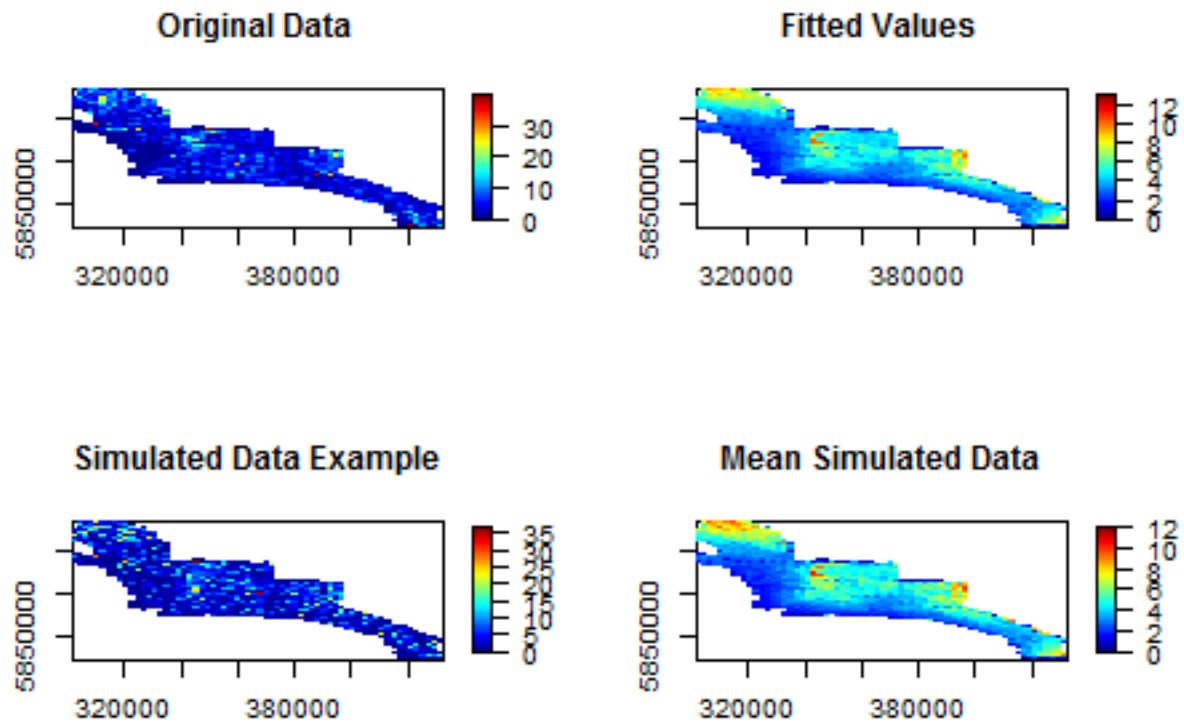
```
plotVariance(bestModel, newdat)
```



```
par(mfrow=c(2,2))
acf(lincs$response, main='Data')
acf(newdat[,1], main='sim data')
acf(residuals(bestModel, type='pearson'), main='original model residuals')
acf(residuals(update(bestModel, newdat[,1] ~ .), type='pearson'), main='sim model residuals')
```



```
par(mfrow=c(2,2))
quilt.plot(lincs$x.pos, lincs$y.pos, lincs$response, main='Original Data')
quilt.plot(lincs$x.pos, lincs$y.pos, fitted(bestModel), main='Fitted Values')
quilt.plot(lincs$x.pos, lincs$y.pos, newdat[,1], main='Simulated Data Example')
quilt.plot(lincs$x.pos, lincs$y.pos, apply(newdat, 1, mean), main='Mean Simulated Data')
```



power analysis

Generate impact data

```
nsim=500
truebeta<-log(0.5)
impdata.auk<-genOverallchangeData(changecoef.link = truebeta, model = bestModel, data = lincs)
```

```
t<-group_by(impdata.auk, eventphase)%>%
  summarise(sum=sum(truth), mean=mean(truth), n=n())
t
```

```
## # A tibble: 2 x 4
##   eventphase      sum      mean      n
##   <dbl>    <dbl>    <dbl> <int>
## 1         0 74355.46  3.940406 18870
## 2         1 37177.73  1.970203 18870
```

```
bef<-impdata.auk[impdata.auk$eventphase==0,]
aft<-impdata.auk[impdata.auk$eventphase==1,]
```

```
spdf1<- SpatialPointsDataFrame( data.frame( x = bef$x.pos , y = bef$y.pos ) , data = data.frame( z = bef$z ) )
e <- extent( spdf1 )
```

```

ratio <- ( e@xmax - e@xmin ) / ( e@ymax - e@ymin )
r <- raster( nrows = 60 , ncols = floor( 60 * ratio ) , ext = extent(spdf1) )
rf <- rasterize( spdf1 , r , field = "z" , fun= mean)
rdf1 <- data.frame( rasterToPoints( rf ), evph=0 )

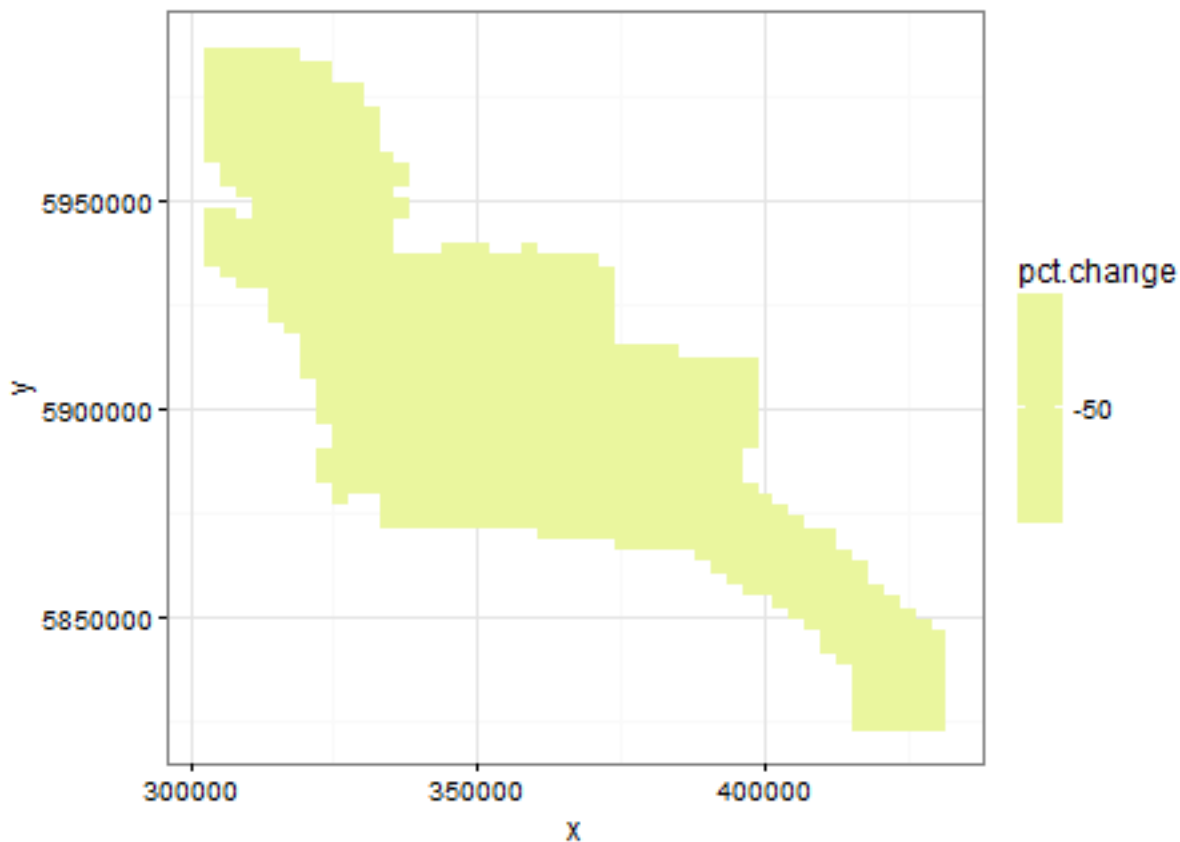
spdf2<- SpatialPointsDataFrame( data.frame( x = aft$x.pos , y = aft$y.pos ) , data = data.frame( z = af
e <- extent( spdf2 )
ratio <- ( e@xmax - e@xmin ) / ( e@ymax - e@ymin )
r <- raster( nrows = 60 , ncols = floor( 60 * ratio ) , ext = extent(spdf2) )
rf <- rasterize( spdf2 , r , field = "z" , fun= mean)
rdf2 <- data.frame( rasterToPoints( rf ), evph=1 )

rdf<-rbind(rdf1, rdf2)
names(rdf)[3]<-'Mean.count'

pct.change<-((rdf$Mean.count[rdf$evph==1] - rdf$Mean.count[rdf$evph==0])/rdf$Mean.count[rdf$evph==0]
)*100

ggplot( NULL ) + geom_raster( data = rdf1 , aes( x , y , fill = pct.change ) ) + scale_fill_gradientn(c

```



```

newdata.auk.imp<-generateNoise(nsim, impdata.auk$truth, family='poisson', d=summary(bestModel)$dispersi

bestModel$splineParams[[1]]$dist<-rbind(bestModel$splineParams[[1]]$dist, bestModel$splineParams[[1]]$d
auksim_glm<-update(bestModel, newdata.auk.imp[,1]~. + eventphase, data=impdata.auk)
auksim_glm$panels<-impdata.auk$panels

```


make sure that the independent data is used to get the null distribution

```
empdistpower.auk<-getRunsCritVals(n.sim = nsim, simData=newdata.auk.imp,  
                                  model = auksim_glm, data = impdata.auk, plot=FALSE, returnDist = TRUE)
```

##

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

```
preddata<-read.csv('../..data/lincspreddata.csv')

# make predictions for month 10 - month with highest estimated density (from month partial plot)
predictdata<-rbind(data.frame(preddata, month=10, area=1, eventphase=0), data.frame(preddata, month=10,
g2k<-makeDists(cbind(predictdata$x.pos, predictdata$y.pos), knotcoords = na.omit(bestModel$splineParams

nsim=200
powerout.auk.oc<-powerSimPl1(newdata.auk.imp, auksim_glm, empdistpower.auk, nsim=nsim, powercoefid=leng

## Code in parallel

save(powerout.auk.oc, file='../FinalReportCode/powerout.lincsauk.oc.RData', compress = 'bzip2')

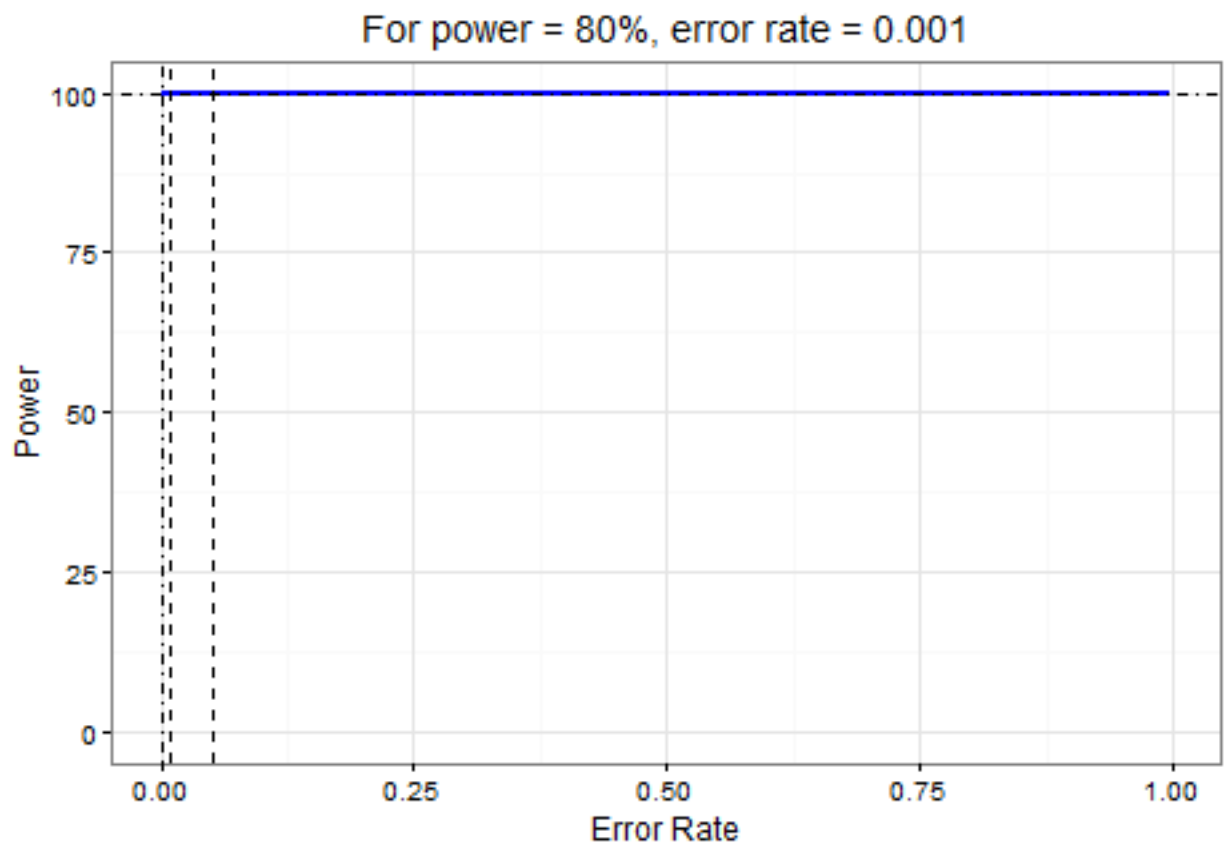
null.output.auk.oc<-pval.coverage.null(newdat.ind = newdata.auk.imp, newdat.corr = NULL, model = auksim.
```

```
save(null.output.auk.oc, file='../FinalReportCode/null.output.lincsauk.oc.RData', compress = 'bzip2')
```

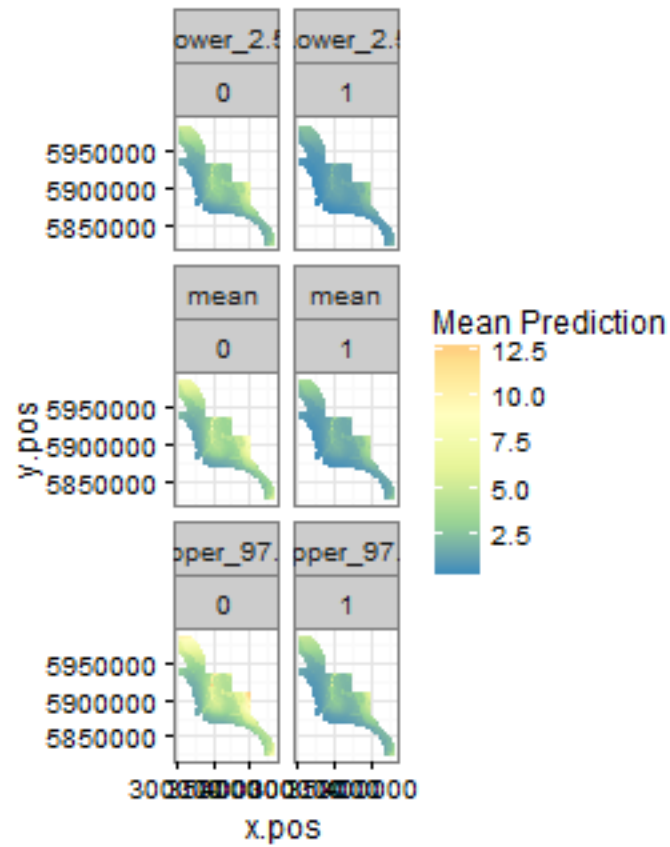
```
summary(powerout.auk.oc, null.output.auk.oc, truebeta=truebeta)
```

```
##
##  ++++ Summary of Power Analysis Output ++++
##
## Number of power simulations = 200
## Number of no change simulations = 100
##
## Power to select 'change' term:
##
##   Under Change (true parameter = -0.6931472) = 100%
##   Under no change (true parameter = 0) = 3%
##
## Coverage for 'change' coefficient:
##
##   Under model = 93.5%
##   Under no change = 96%
##
## Overall Abundance Summary with 95% Confidence Intervals:
##
##      Abundance LowerCI UpperCI
## Before  5300225 4547702 6117703
## After   2656925 2266837 3104230
##
## Note: These calculations assume the correct area has been given for each prediction grid cell.
```

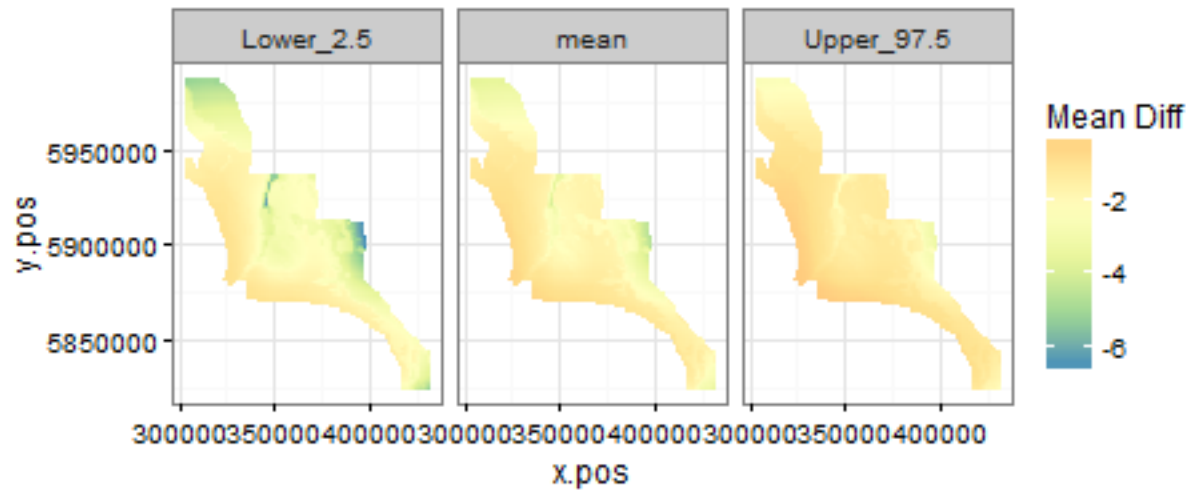
```
powerPlot(powerout.auk.oc)
```



```
plot.preds(powerout.auk.oc)
```

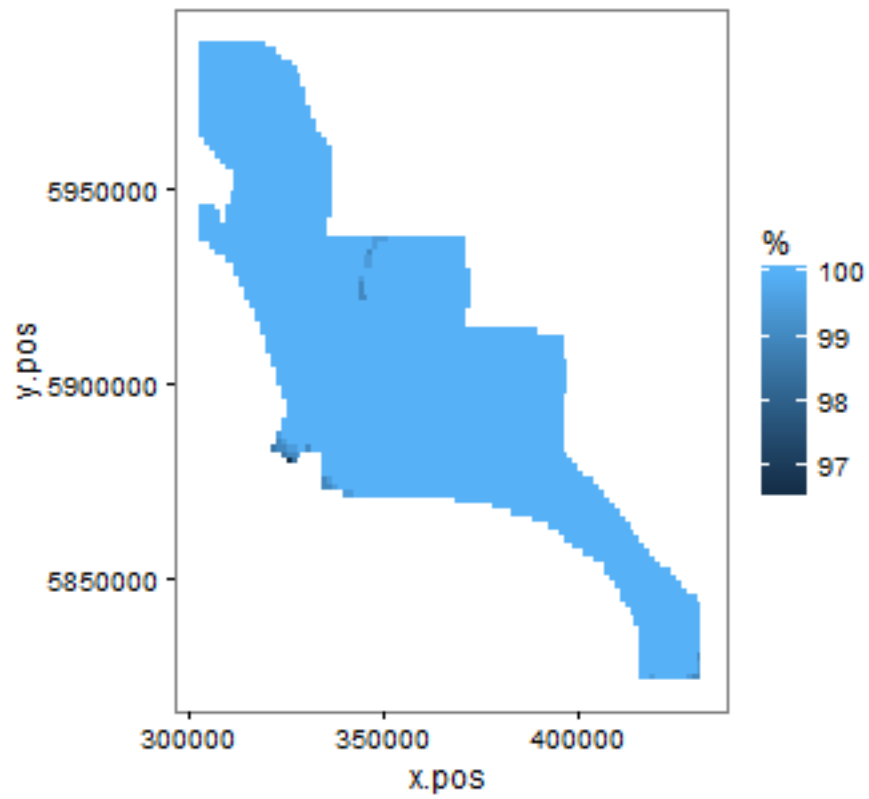


```
plot.diffs(powerout.auk.oc)
```



```
plot.sigdiff(powerout.auk.oc, coordinates = predictdata[predictdata$eventphase==0,c('x.pos', 'y.pos')],
```

Individual 5% Error Rate, Adj = "none"



```
plot.sigdiff(powerout.auk.oc, coordinates = predictdata[predictdata$eventphase==0,c('x.pos', 'y.pos')],
```


Family ER = 5%; Invididual ER = 0% , Adj = "sidak"

