Likelihood analysis of spatial capture-recapture models

Spatial Capture-Recapture Workshop,

Cornell University, Ithaca, NY

April 2014

Bayesian analysis of SCR models

- So far we have seen how to fit fully spatial capture recapture models in:

  $\boldsymbol{SCR_0}$: most basic form of an SCR model [`bear.fit1j`]

- We also fitted a series of models to account for additional detection heterogeneity:

  $\boldsymbol{SCR_{trap}}$: trap level response, e.g. baited/non-baited [`bear.fit2j`]

  $\boldsymbol{SCR_b}$: behavioral response, e.g. trap-happiness/trap-shyness [`bear.fit3j`]

  $\boldsymbol{SCR_{s,b}}$: sex effect (p_0) and behavioral response [`bear.fit4j`]

Likelihood analysis

- These models can also be analyzed using likelihood methods
- Likelihood of the encounter data, conditional on activity center, $\boldsymbol{s}_i$, is

$$L(y_i|\boldsymbol{s}_i) = \prod_{j=1}^{J} \prod_{k=1}^{K} Bernoulli(y_{ijk}|p(x_j, s_i))$$

- But, we don't know $\boldsymbol{s}_i$, it is a **latent variable**/a **random effect**
- We can analyze this two ways:
  - Put a prior distribution on it and analyze by **MCMC**
  - Put a prior distribution in it and analyze it by **integrated likelihood**, in which we remove the random effect from the conditional distribution by integration

Likelihood analysis

- These models can also be analyzed using likelihood methods
- Conditional-on-s likelihood:

$$L(y_i|\boldsymbol{s}_i) = \prod_{j=1}^{J} \prod_{k=1}^{K} Bernoulli(y_{ijk}|p(x_j, s_i))$$

- But, we don't know $\boldsymbol{s}_i$, it is a **latent variable**/a **random effect**
- Remove the random effect by integration (over discrete space!)
- Integrated likelihood:

$$L(y_i) = \int_S \prod_{j=1}^{J} \prod_{k=1}^{K} Bernoulli(y_{ijk}|p(x_j, s_i)) \; \boldsymbol{ds}$$

- Doesn't depend on **s** anymore and can be maximized!

Doing likelihood analysis

1. Use R package `secr`

2. Write an R function that does this integration. Sometimes this comes in handy

$$L(y_i) = \int_S \prod_{j=1}^{J} \prod_{k=1}^{K} Bernoulli(y_{ijk} | p(x_j, s_i)) \; \boldsymbol{ds}$$

I have purposely omitted some key technical details regarding how N and D are estimated. Ch 6 of the SCR book gives an much more technical overview of likelihood analysis of spatial capture-recapture models!

Likelihood analysis of SCR models – '`secr`'

- Many of you may have heard of `secr` – 'spatially explicit capture recapture'
- R package developed and maintained by Murray Efford

- This is ***NOT*** a workshop on how to use `secr`
- `secr` is, however, very useful and worth knowing about
- A ***brief*** overview of the package can be found in the SCR book (again see Ch. 6)
- This website has everything you need to know about `secr` (and `DENSITY`):

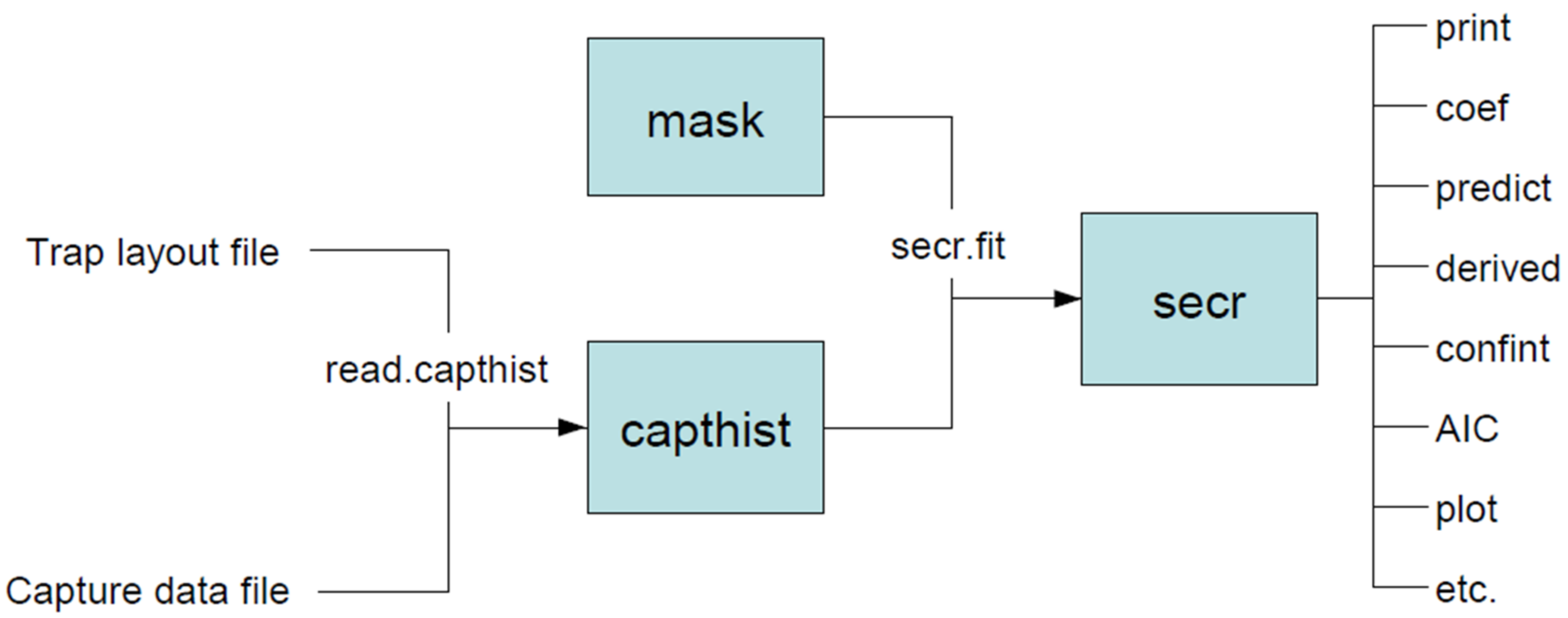
http://www.otago.ac.nz/density/SECR.html

Likelihood analysis of SCR models – ‘`secr`’

- `secr` – ‘spatially explicit capture recapture’
- R package developed and maintained by Murray Efford
- Very convenient and user-friendly package:
    - Data manipulation
    - Model fitting by maximum likelihood
    - AIC-based model selection & model averaging
    - Simulation
    - Example datasets
- Maximum likelihood is appealing:
    - Faster than BUGS analysis
    - Straight forward model selection
    - Efficient for LARGE problems

Analysis using `secr`

- A typical `secr` workflow



Efford (2013) - http://www.otago.ac.nz/density/pdfs/secr-overview.pdf

Analysis using `secr`

- A typical `secr` workflow
    1. Load the relevant libraries and data

```
# load the scrpart1.Rdata workspace (double click it!)
library("secr")
library("scrbook")
library("coda")
data("beardata")
```

Analysis using `secr`

`secr function!`

- A typical `secr` workflow
    1. Load the relevant libraries and data
    2. Trap layout file ‘`trapfile`’

```
# Make the trap file
traps <- as.matrix(cbind(c(1:dim(beardata$trapmat)[1]), beardata$trapmat*1000))
colnames(traps) <- c("trapID", "x", "y")
set.seed(2013)
lure <- rbinom(nrow(traps),1,.5)
traps1 <- as.data.frame(traps[, 1:3])
trapfile <- read.traps(data = traps1, detector = "proximity")
covariates(trapfile) <- data.frame(lure=lure)

plot(trapfile)
head(traps1)
plot(traps1[,-1],pch=3,lwd=2)
text(traps1[,2]+500,traps1[,3],paste(traps1[,1]),col="grey")
```

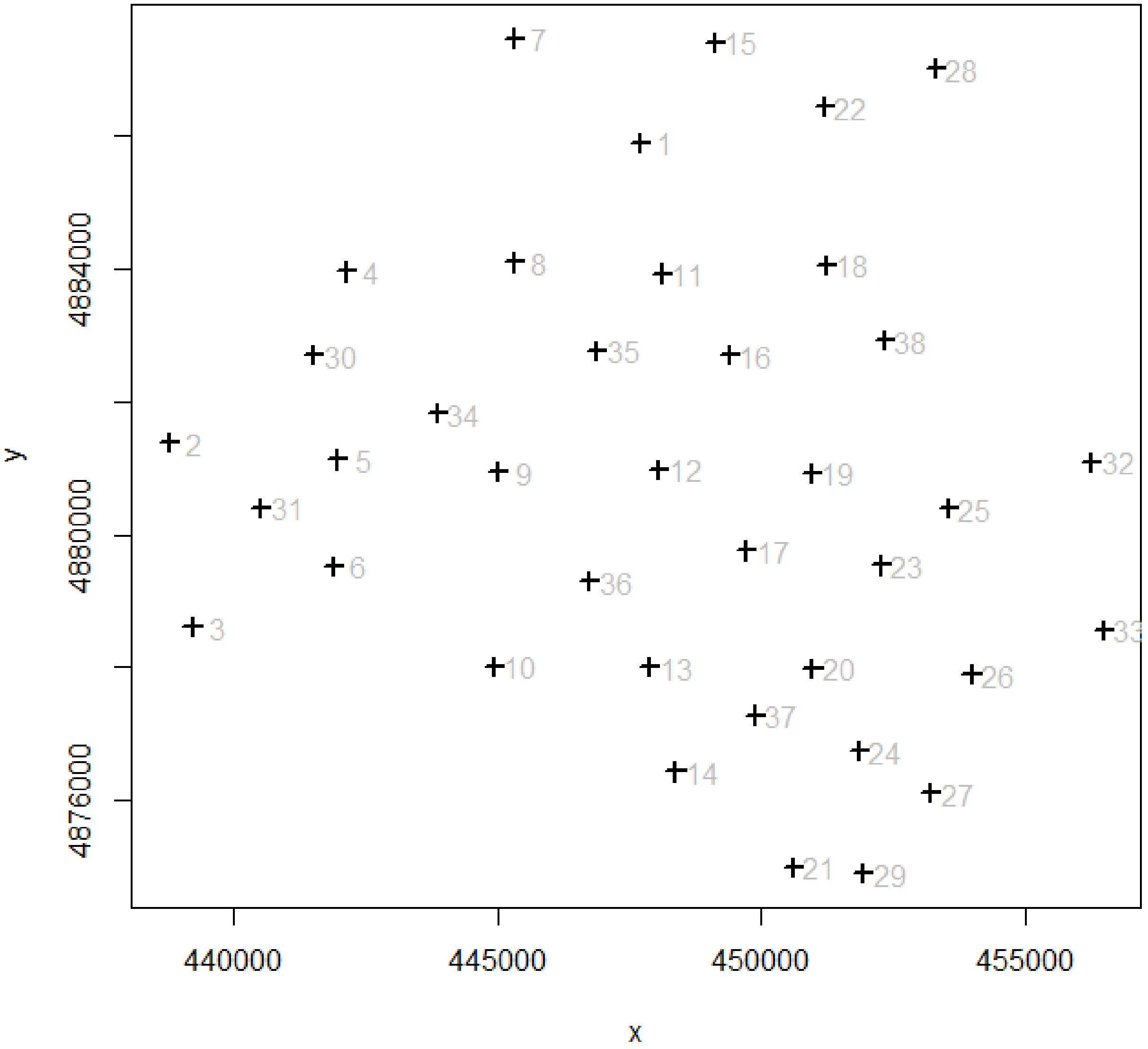
Detector types -> encounter model

- Count -> Poisson encounter model
  - Detections independent between detectors and more than 1 allowed per detector
  - e.g., camera trapping
- Proximity detector -> Binomial encounter model
  - Detections independent between detectors
  - e.g., acoustic surveys, **hair snares**
- Multi-catch trap -> Multinomial encounter model
  - Detectors "compete" for animals
  - e.g., pitfall traps
- Single-catch trap -> ? Currently no model for this type
  - Detectors compete and fill up
  - Only available for simulation right now
  - e.g., Sherman traps

Analysis using `secr`

- A typical `secr` workflow
    1. Load the relevant libraries and data
    2. Trap layout file ‘`trapfile`’

```
> head(traps1)
  trapID      x       y
1      1 447694 4885900
2      2 438789 4881388
3      3 439232 4878611
4      4 442142 4883961
5      5 441987 4881130
6      6 441900 4879523
```



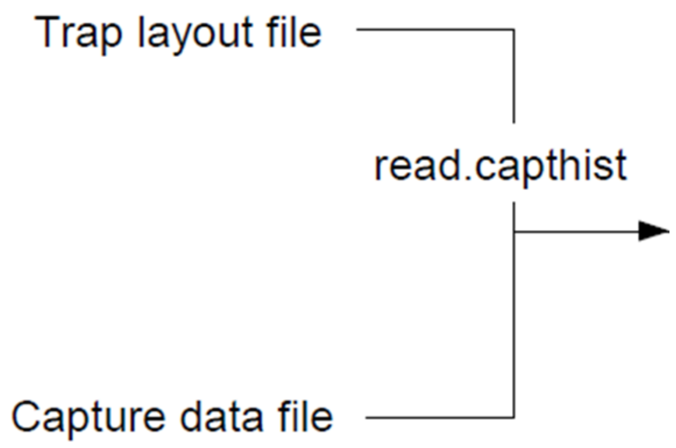
Analysis using `secr`

- A typical `secr` workflow
  1. Load the relevant libraries and data
  2. Trap layout file ‘`trapfile`’
  3. Capture data file ‘`capfile`’

```
capfile <- as.data.frame(beardata$flat)
head(capfile)
```

```
> head(capfile)
  Session ID Occasion trapID
1       1  1        1      2
2       1  1        3      5
3       1  1        5      5
4       1  1        5     31
5       1  1        5     34
6       1  1        6      5
```

Analysis using `secr`



- A typical `secr` workflow
    1. Load the relevant libraries and data
    2. Trap layout file ‘`trapfile`’
    3. Capture data file ‘`capfile`’

```
bear.cap <- make.capthist(capfile,          # capture file
                          trapfile,         # trap file
                          fmt = "trapID",   # trap data format
                          noccasions = 8)   # number of occasions
```

THIS IS OUR MODEL OBJECT!!

fmt = "XY“: the required fields are (session, ID, occasion, x, y)

fmt ="trapID“: the required fields are (session, ID, occasion, trap)

Analysis using secr

$\boldsymbol{SCR_0}$: most basic form of an SCR model [bear.fit1j]

```
bear.0 <-secr.fit(bear.cap,
                  model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
                  detectfn = NULL,
                  buffer = 20000)
```

Analysis using `secr`

$\boldsymbol{SCR_0}$: most basic form of an SCR model [`bear.fit1j`]

```
bear.0 <-secr.fit(bear.cap,
                  model = list(D ~ 1, g0 ~ 1, sigma ~ 1),
                  detectfn = NULL,
                  buffer = 20000)
```

Here we specify the model we wish to fit, i.e. SCR_0:

```
list(D ~ 1, g0 ~ 1, sigma ~ 1)
```

- `D ~ 1` → Density Model
- `g0 ~ 1` → Baseline Encounter Probability
- `sigma ~ 1` → Scale parameter (movement)

Analysis using `secr`: ‘`detectfn =`‘

Some encounter models and their corresponding model notation.

Code	Name	Parameters	Function
0 HN	halfnormal	g0, sigma	$g(d) = g_0 \exp\left(\frac{-d^2}{2\sigma^2}\right)$
1 HR	hazard rate	g0, sigma, z	$g(d) = g_0[1 - \exp\{-({}^d/_\sigma)^{-z}\}]$
2 EX	exponential	g0, sigma	$g(d) = g_0 \exp\{-({}^d/_\sigma)\}$
3 CHN	compound halfnormal	g0, sigma, z	$g(d) = g_0[1 - \{1 - \exp\left(\frac{-d^2}{2\sigma^2}\right)\}^z]$
4 UN	uniform	g0, sigma	$g(d) = g_0, d <= \sigma; g(d) = 0$, otherwise
5 WEX	w exponential	g0, sigma, w	$g(d) = g_0, d < w; g(d) = g_0 \exp\left(-\frac{d-w}{\sigma}\right)$, otherwise
6 ANN	annular normal	g0, sigma, w	$g(d) = g_0 \exp\{\frac{-(d-w)^2}{2\sigma^2}\}$
7 CLN	cumulative lognormal	g0, sigma, z	$g(d) = g_0[1 - F\{(d - \mu)/s\}]$
8 CG	cumulative gamma	g0, sigma, z	$g(d) = g_0\{1 - G(d; k, \theta)\}$
9 BSS	binary signal strength	b0, b1	$g(d) = 1 - F\{-(b_0 + b_1 d)\}$
10 SS	signal strength	beta0, beta1, sdS	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1 d)\}/s]$
11 SSS	signal strength spherical	beta0, beta1, sdS	$g(d) = 1 - F[\{c - (\beta_0 + \beta_1(d-1) - 10\log_{10} d^2)\}/s]$
14 HHN	hazard halfnormal	lambda0, sigma	$\lambda(d) = \lambda_0 \exp\left(\frac{-d^2}{2\sigma^2}\right); g(d) = 1 - \exp(-\lambda(d))$
15 HHR	hazard hazard rate	lambda0, sigma, z	$\lambda(d) = \lambda_0(1 - \exp\{-({}^d/_\sigma)^{-z}\}); g(d) = 1 - \exp(-\lambda(d))$
16 HEX	hazard exponential	lambda0, sigma	$\lambda(d) = \lambda_0 \exp\{-({}^d/_\sigma)\}; g(d) = 1 - \exp(-\lambda(d))$
17 HAN	hazard annular normal	lambda0, sigma, w	$\lambda(d) = \lambda_0 \exp\{\frac{-(d-w)^2}{2\sigma^2}\}; g(d) = 1 - \exp(-\lambda(d))$
18 HCG	hazard cumulative gamma	lambda0, sigma, z	$\lambda(d) = \lambda_0\{1 - G(d; k, \theta)\}; g(d) = 1 - \exp(-\lambda(d))$

Efford (2013) - http://www.otago.ac.nz/density/pdfs/secr-overview.pdf

Analysis using `secr`

$\boldsymbol{SCR_0}$: most basic form of an SCR model [`bear.fit1j`]

```
bear.0 = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=20000)
```

```
> comp
          D     p0 sigma
secr  0.166 0.106 1.973
bayes 0.174 0.109 1.959
```

Analysis using `secr`

Some built-in models and their corresponding model notation for some of the most commonly used (S)CR models.

Variable	Description	Notes
g	group	interaction of the capthist individual covariates listed in argument `groups` of `secr.fit`
t	time factor	one level for each occasion
T	time trend	linear trend over occasions on link scale
b	learned response	step change after first detection
B	transient response	depends on detection at preceding occasion (Markovian response)
bk	animal x site response	site-specific step change
Bk	animal x site response	site-specific transient response
k	site learned response	site effectiveness changes once any animal caught
K	site transient response	site effectiveness depends on preceding occasion
session	session factor	one level for each session
Session	session trend	linear trend on link scale
h2	2-class mixture	finite mixture model with 2 latent classes

Efford (2013) - http://www.otago.ac.nz/density/pdfs/secr-overview.pdf

Analysis using `secr`

$\boldsymbol{SCR_0}$: most basic form of an SCR model [`bear.fit1j`]

```
bear.0 = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=20000)
```

$\boldsymbol{SCR_{trap}}$: trap level response, e.g. baited/non-baited [`bear.fit2j`]

```
bear.trap = secr.fit(bear.cap, model = list(D ~ 1, g0 ~ lure, sigma ~ 1),…)
```

Analysis using `secr`

$\boldsymbol{SCR_0}$: most basic form of an SCR model [`bear.fit1j`]

```
bear.0 = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=20000)
```

$\boldsymbol{SCR_{trap}}$: trap level response, e.g. baited/non-baited [`bear.fit2j`]

```
bear.trap = secr.fit(bear.cap, model = list(D ~ 1, g0 ~ lure, sigma ~ 1),…)
```

$\boldsymbol{SCR_b}$: behavioral response, e.g. trap-happiness/trap-shyness [`bear.fit3j`]

```
bear.b = secr.fit(bear.cap, model = list(D ~ 1, g0 ~ b, sigma ~ 1),…)
```

Analysis using `secr`

$\boldsymbol{SCR_0}$: most basic form of an SCR model [`bear.fit1j`]

```
bear.0 = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1), buffer=20000)
```

$\boldsymbol{SCR_{trap}}$: trap level response, e.g. baited/non-baited [`bear.fit2j`]

```
bear.trap = secr.fit(bear.cap, model = list(D ~ 1, g0 ~ lure, sigma ~ 1),…)
```

$\boldsymbol{SCR_b}$: behavioral response, e.g. trap-happiness/trap-shyness [`bear.fit3j`]

```
bear.b = secr.fit(bear.cap, model = list(D ~ 1, g0 ~ b, sigma ~ 1),…)
```

$\boldsymbol{SCR_{s,b}}$: sex effect (p_0) and behavioral response [`bear.fit4j`]

```
bear.bsex = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b + session, sigma ~ 1),…)
```

Analysis using `secr` – AIC and model selection

- ML allows for multi-model inference, e.g. AIC based model selection/averaging
- Make AIC model tables (`bear.fit5j`)

```
bear.0 = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ 1, sigma ~ 1))
bear.b = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b , sigma ~ 1))
bear.sex = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ session, sigma ~ 1))
bear.bsex = secr.fit(bear.cap, model=list(D ~ 1, g0 ~ b + session, sigma ~ 1))
```

```
AIC.tab <- AIC(bear.0, bear.b, bear.sex, bear.bsex)
```

```
> (AIC.tab <- AIC(bear.0, bear.b, bear.sex, bear.bsex))
                                    model   detectfn npar    logLik      AIC     AICc  dAICc AICcwt
bear.bsex D~1 g0~b + session sigma~1 halfnormal    5 -628.6209 1267.242 1268.705  0.000 0.6153
bear.b               D~1 g0~b sigma~1 halfnormal    4 -630.3460 1268.692 1269.644  0.939 0.3847
bear.sex       D~1 g0~session sigma~1 halfnormal    4 -635.1545 1278.309 1279.261 10.556 0.0000
bear.0               D~1 g0~1 sigma~1 halfnormal    3 -636.5237 1279.047 1279.606 10.901 0.0000
```

Analysis using `secr` –model averaging

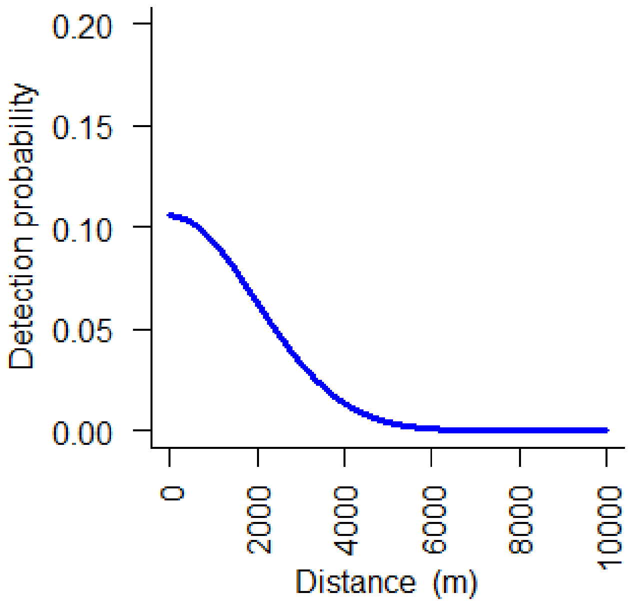
- ML allows for multi-model inference, e.g. AIC based model selection/averaging
- Calculate AIC weighted model averaged parameter estimates:

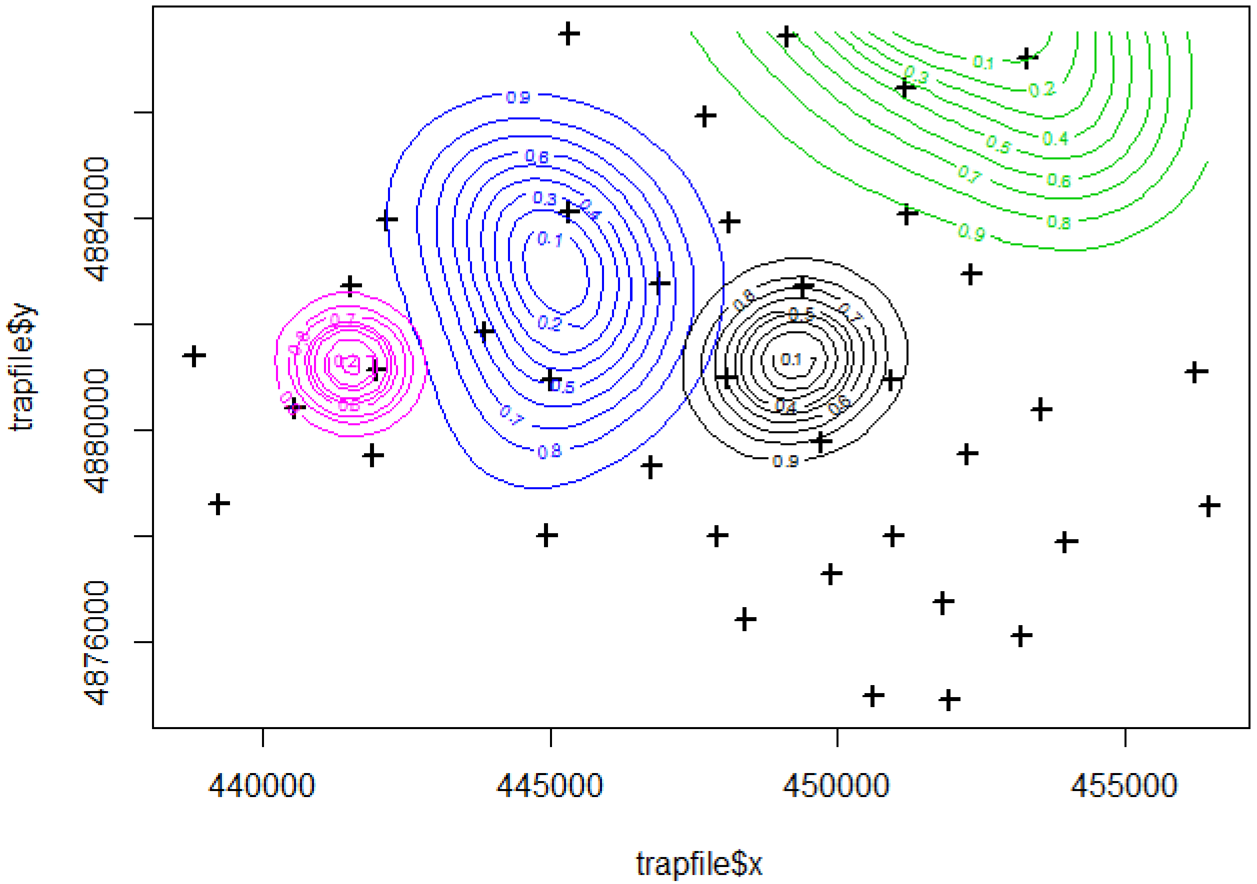
```
mod.avg <- model.average(bear.0, bear.trap, bear.b, bear.bsex)
```

```
> mod.avg
, , D
                 estimate SE.estimate       lcl      ucl
session=1,b=0 0.2265037  0.03966843 0.1611112 0.318438
session=2,b=0 0.2265037  0.03966843 0.1611112 0.318438
, , g0
                  estimate SE.estimate        lcl        ucl
session=1,b=0 0.03664258  0.01035559 0.02095523 0.06331423
session=2,b=0 0.04593377  0.01269371 0.02656486 0.07828911
, , sigma
                estimate SE.estimate      lcl      ucl
session=1,b=0 2339.674    137.0489 2086.114 2624.053
session=2,b=0 2339.674    137.0489 2086.114 2624.053
```

Analysis using `secr` – various plotting functions

```
# det function
plot(bear.0, xval=0:10000, ylim=c(0,0.2), lwd=3, col=4, bty="l", las=2)
# hr centers w/ confidence contours
plot(trapfile$x,trapfile$y)
fxi.contour(bear.0,i=7,add=T,col=4)
fxi.contour(bear.0,i=8,add=T,col=1)
fxi.contour(bear.0,i=15,add=T,col=3)
fxi.contour(bear.0,i=1,add=T,col=6)
```





Analysis using `secr` – so why not use it for everything?

http://www.otago.ac.nz/density/pdfs/secr-manual.pdf

- Less transparent than the Bayesian (BUGS) analyses we have seen
- Limited to models currently available in `secr` – which, admittedly, is a lot
- Sometimes we require the flexibility of writing our own code/models

- `scrbook`, the R package that accompanies the book, has a suit of integrated likelihood functions called `intlikX` that can be, edited, manipulated and taylored to suit specific needs

See `intlik4` function – in editor!