

Spatial Capture Recapture Data Analysis

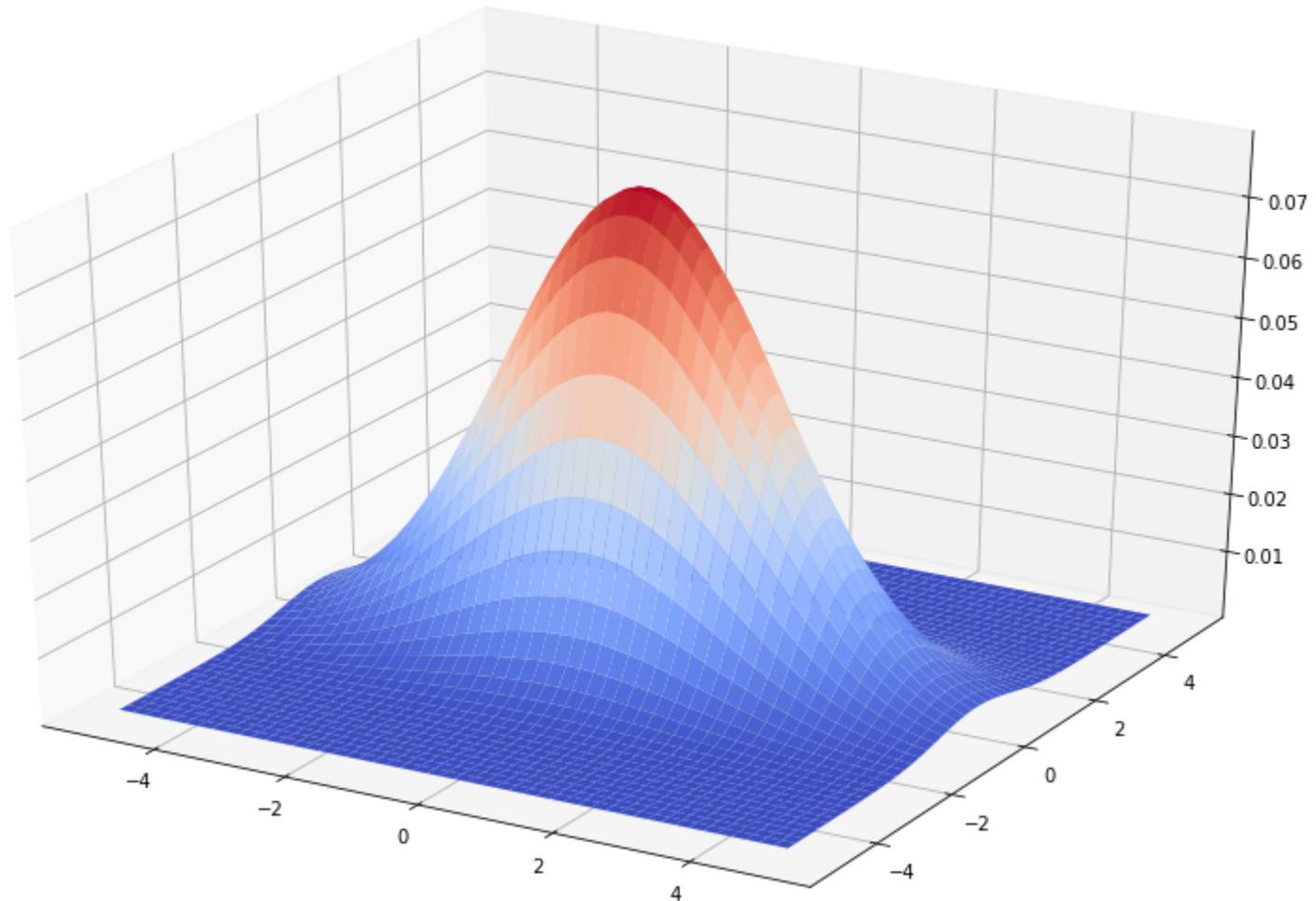
Koustubh Sharma & Justine Shanti Alexander



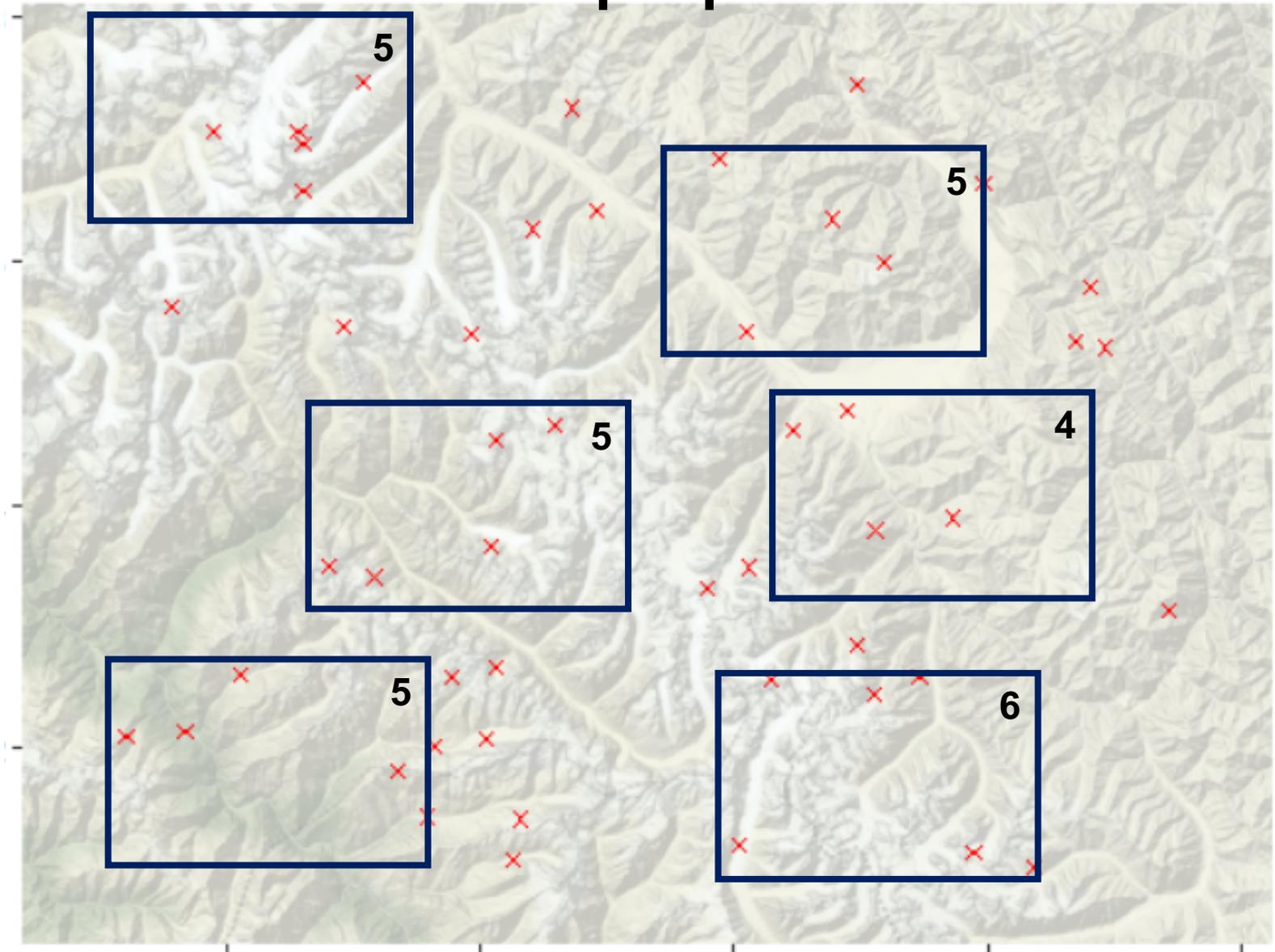
SNOW
LEOPARD
NETWORK



Maximum Likelihood

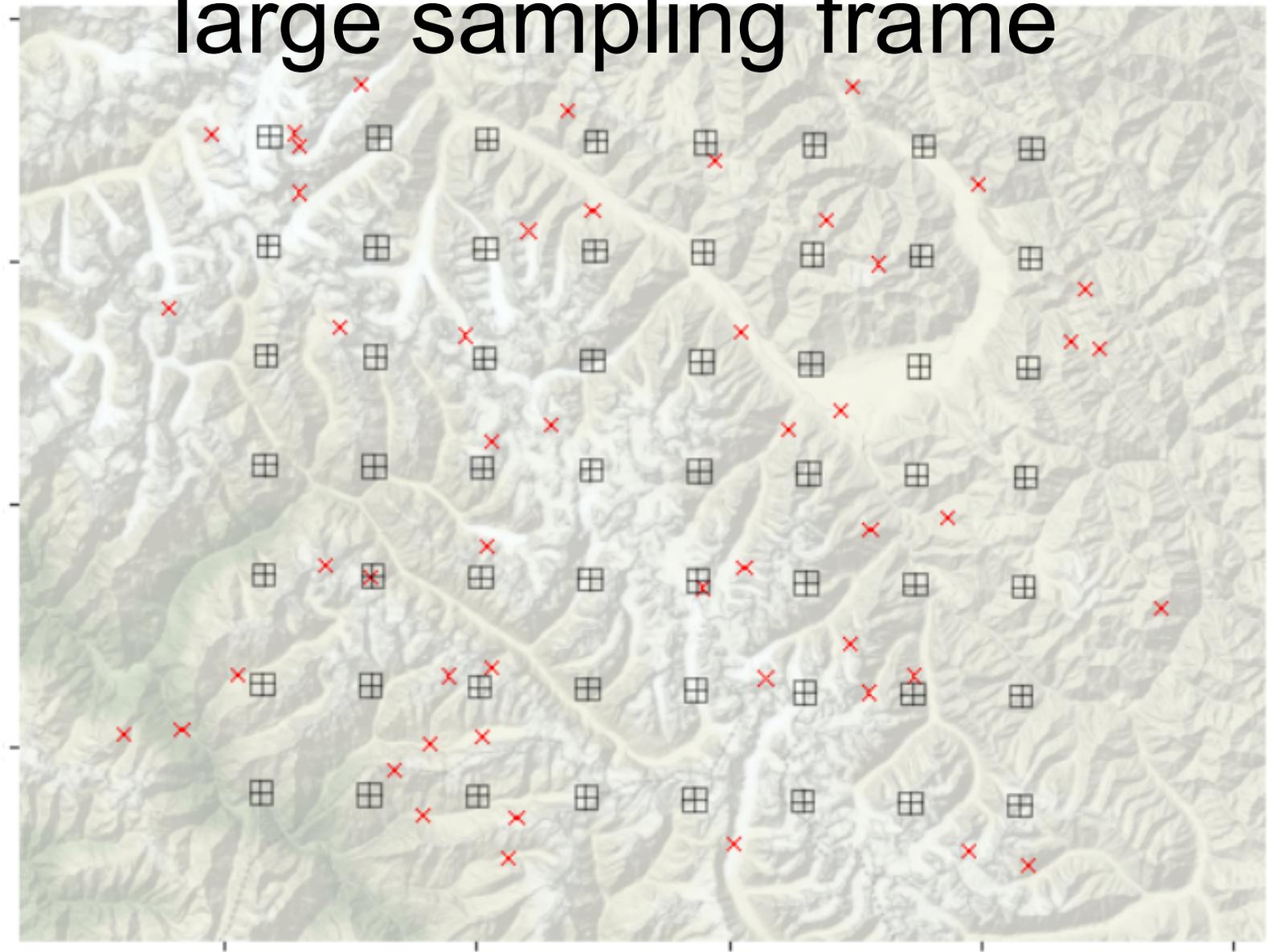


Random population



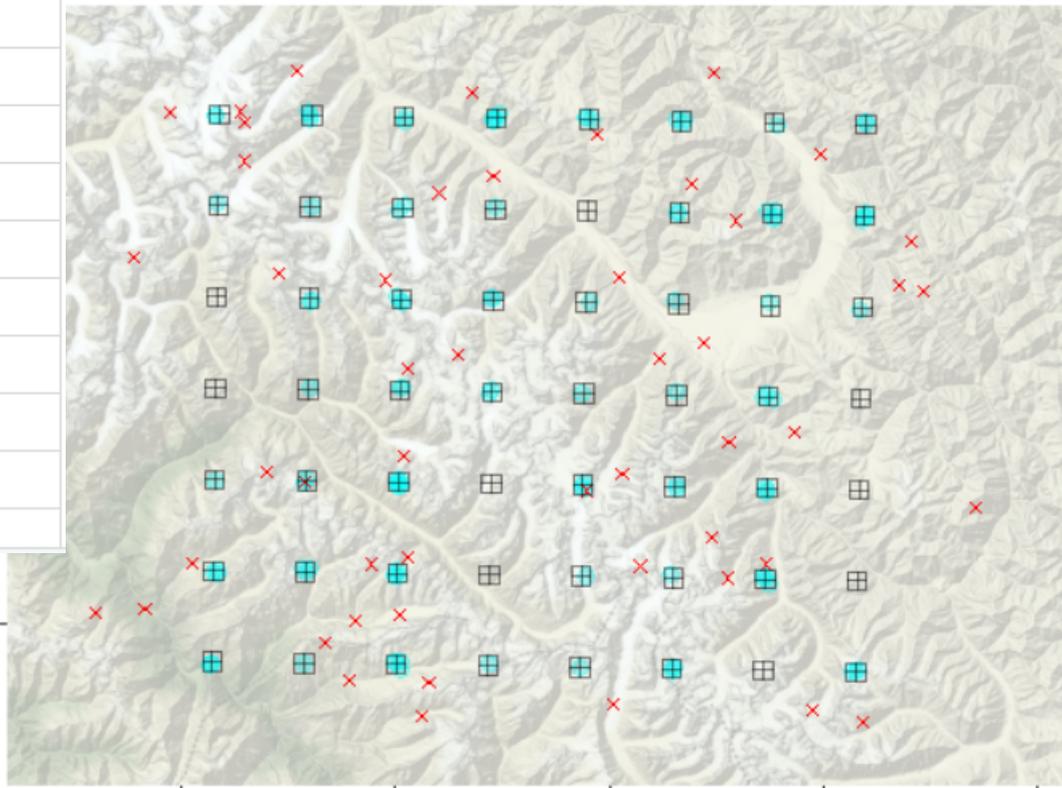
Scenario 1:

large sampling frame



Capture data

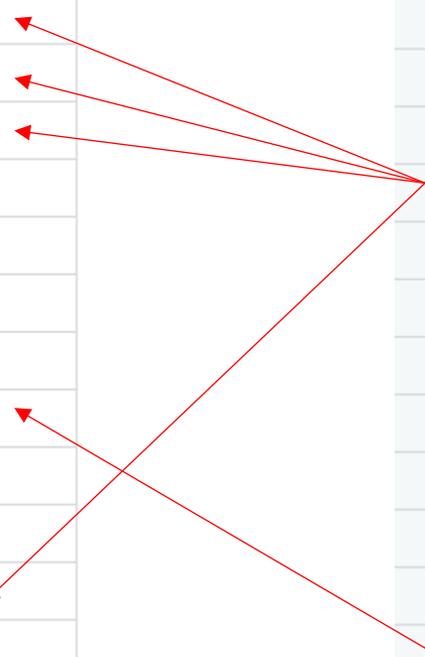
	Session	ID	Occasion	trapID
1	1	2	1	C15
2	1	2	1	C15
3	1	2	1	C15
4	1	3	1	C37
5	1	3	1	C37
6	1	3	1	C37
7	1	3	1	C37
8	1	5	1	C23
9	1	6	1	C40
10	1	8	1	C51
11	1	9	1	C15
12	1	10	1	C28
13	1	10	1	C27



Trap data

	Session	ID	Occasion	trapID
1	1	2	1	C15
2	1	2	1	C15
3	1	2	1	C15
4	1	3	1	C37
5	1	3	1	C37
6	1	3	1	C37
7	1	3	1	C37
8	1	5	1	C23
9	1	6	1	C40
10	1	8	1	C51
11	1	9	1	C15
12	1	10	1	C28
13	1	10	1	C27

	x	y
C12	650000	3680000
C13	660000	3680000
C14	670000	3680000
C15	680000	3680000
C16	690000	3680000
C17	620000	3690000
C18	630000	3690000
C19	640000	3690000
C20	650000	3690000
C21	660000	3690000
C22	670000	3690000
C23	680000	3690000
C24	690000	3690000



R for data analysis

<https://cran.r-project.org/bin/windows/base/>



<https://rstudio.com/products/rstudio/download/>



R Studio Workspace

The screenshot shows the R Studio interface with several components highlighted by green circles:

- Source Editor:** Contains R code for installing and loading the 'secr' package:

```
1 install.packages("secr")
2 library(secr)
```
- Environment Pane:** Lists objects in the Global Environment:

Object	Type
p_anim2	List of 14
partmat	2635 obs. of 1 variable
pbar	List of 3
plot	List of 9
plot.hist.ani...	List of 14
plot.many	List of 9
plot.many.a	List of 14
plot.normal	List of 9
plot.normal.a	List of 14
- Console:** Shows the execution of the code and resulting messages:

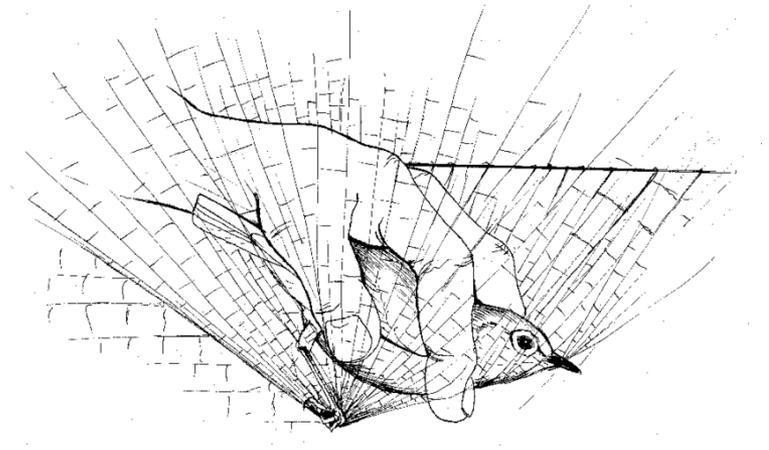
```
[Workspace loaded from C:/Users/koust/Dropbox (Snow Leopard Trust)/CREEM/Analyses/SouthGobi/Spiti_Rishi/.RData]
Loading required package: sp
> Spiti.pop.large.3.5
Error: object 'Spiti.pop.large.3.5' not found
> library(secr)
This is secr 4.2.2. For overview type ?secr
Warning message:
package 'secr' was built under R version 3.6.3
>
```

Prepare R

```
install.packages("secr")  
library(secr)
```

Traps revisited

- Single
- Multi
- Proximity
- Count
- Capped
- Polygon
- Transect
- Signal
- Telemetry



Capture and Trap data

large.cap

	Session	ID	Occasion	trapID
1	1	2	1	C15
2	1	2	1	C15
3	1	2	1	C15
4	1	3	1	C37
5	1	3	1	C37
6	1	3	1	C37
7	1	3	1	C37
8	1	5	1	C23
9	1	6	1	C40
10	1	8	1	C51
11	1	9	1	C15
12	1	10	1	C28
13	1	10	1	C27

large.trap

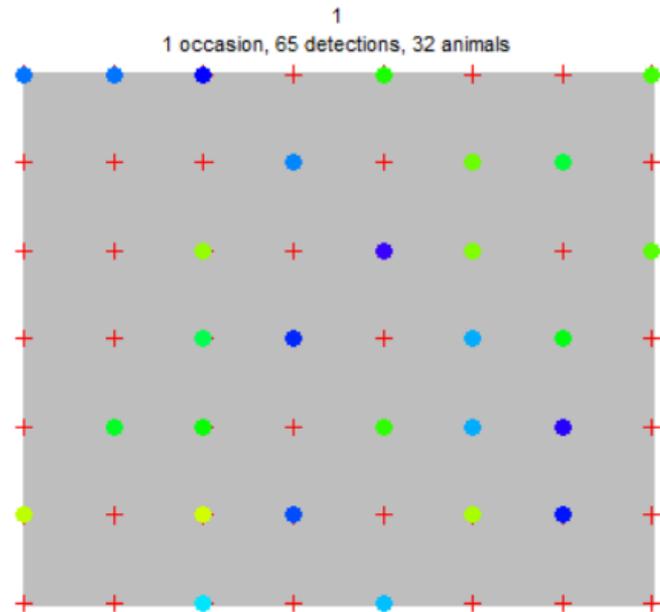
	x	y
C12	650000	3680000
C13	660000	3680000
C14	670000	3680000
C15	680000	3680000
C16	690000	3680000
C17	620000	3690000
C18	630000	3690000
C19	640000	3690000
C20	650000	3690000
C21	660000	3690000
C22	670000	3690000
C23	680000	3690000
C24	690000	3690000

```
Large.cap<-read.csv("large.cap.csv")
```

```
large.trap<-read.traps(data=large.trap, detector = 'count')
```

Input data in R

```
all.data.large<-  
make.caphist(captures = large.cap,  
traps = large.trap, fmt="trapID")
```

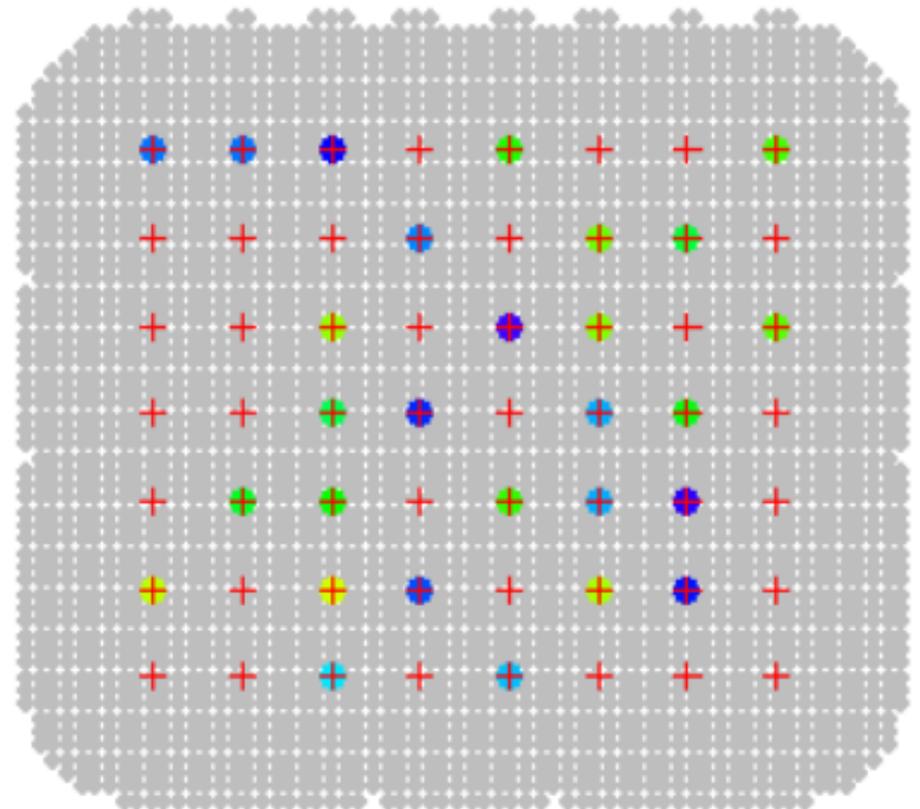


Masks!

Create a mask (integration area)

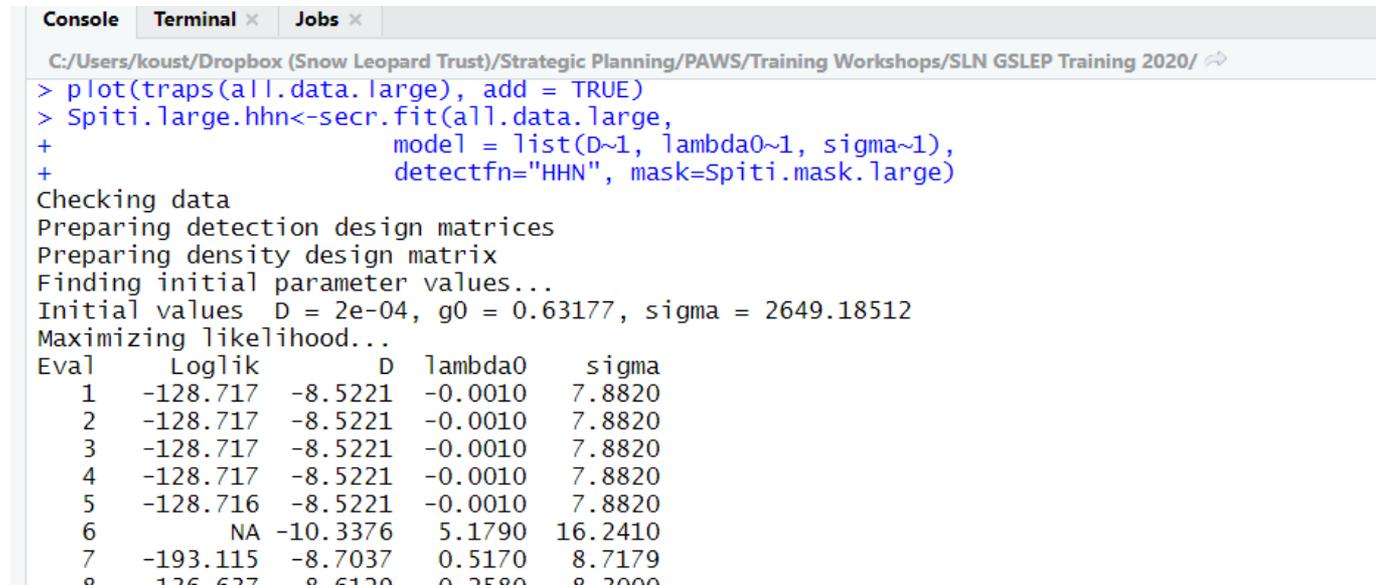
```
Spiti.mask.large<-  
make.mask(traps(all.data.large),  
buffer=15000,  
type="trapbuffer")
```

1 occasion, 65 detections, 32 animals



Run your first SCR analysis

```
Spiti.large.hhn<-  
secr.fit(all.data.large,  
model= list(D~1, lambda0~1, sigma~1),  
detectfn="HHN",  
mask=Spiti.mask.large)
```



```
Console Terminal x Jobs x  
C:/Users/koust/Dropbox (Snow Leopard Trust)/Strategic Planning/PAWS/Training Workshops/SLN GSLEP Training 2020/ ↗  
> plot(traps(all.data.large), add = TRUE)  
> Spiti.large.hhn<-secr.fit(all.data.large,  
+ model = list(D~1, lambda0~1, sigma~1),  
+ detectfn="HHN", mask=Spiti.mask.large)  
Checking data  
Preparing detection design matrices  
Preparing density design matrix  
Finding initial parameter values...  
Initial values D = 2e-04, g0 = 0.63177, sigma = 2649.18512  
Maximizing likelihood...  
Eval Loglik D lambda0 sigma  
1 -128.717 -8.5221 -0.0010 7.8820  
2 -128.717 -8.5221 -0.0010 7.8820  
3 -128.717 -8.5221 -0.0010 7.8820  
4 -128.717 -8.5221 -0.0010 7.8820  
5 -128.716 -8.5221 -0.0010 7.8820  
6 NA -10.3376 5.1790 16.2410  
7 -193.115 -8.7037 0.5170 8.7179  
8 126.637 8.6120 0.2580 8.3000
```

SCR analyses take time!

```
Console Terminal x Jobs x
C:/Users/koust/Dropbox (Snow Leopard Trust)/Strategic Planning/PAWS/Training Workshops/SLN GSLEP Training 2020/ ↗
52 -112.058 -9.5094 1.0439 7.9848
53 -112.058 -9.5094 1.0439 7.9848
54 -112.058 -9.5094 1.0439 7.9848
55 -112.058 -9.5094 1.0439 7.9848
56 -112.058 -9.5094 1.0439 7.9848
57 -112.058 -9.5094 1.0439 7.9848
58 -112.058 -9.5093 1.0439 7.9848
59 -112.058 -9.5094 1.0440 7.9848
60 -112.058 -9.5094 1.0439 7.9856
61 -112.058 -9.5092 1.0439 7.9848
62 -112.058 -9.5093 1.0440 7.9848
63 -112.058 -9.5093 1.0439 7.9856
64 -112.058 -9.5094 1.0441 7.9848
65 -112.058 -9.5094 1.0440 7.9856
66 -112.059 -9.5094 1.0439 7.9864
Completed in 6.62 seconds at 10:39:54 27 Jul 2020
>
```

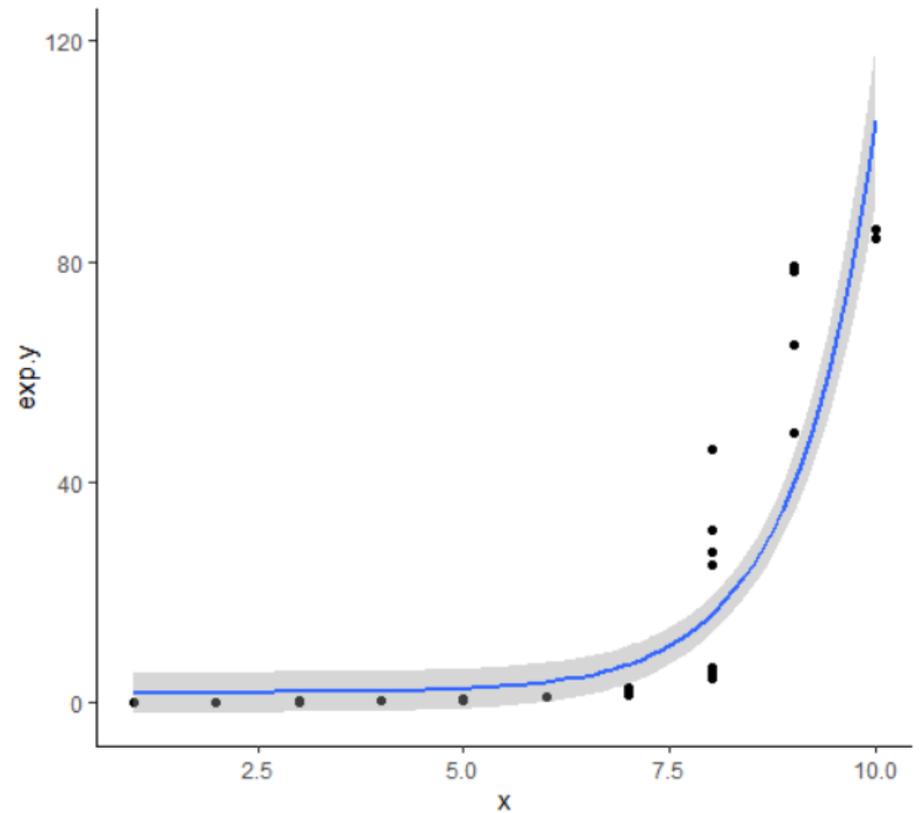
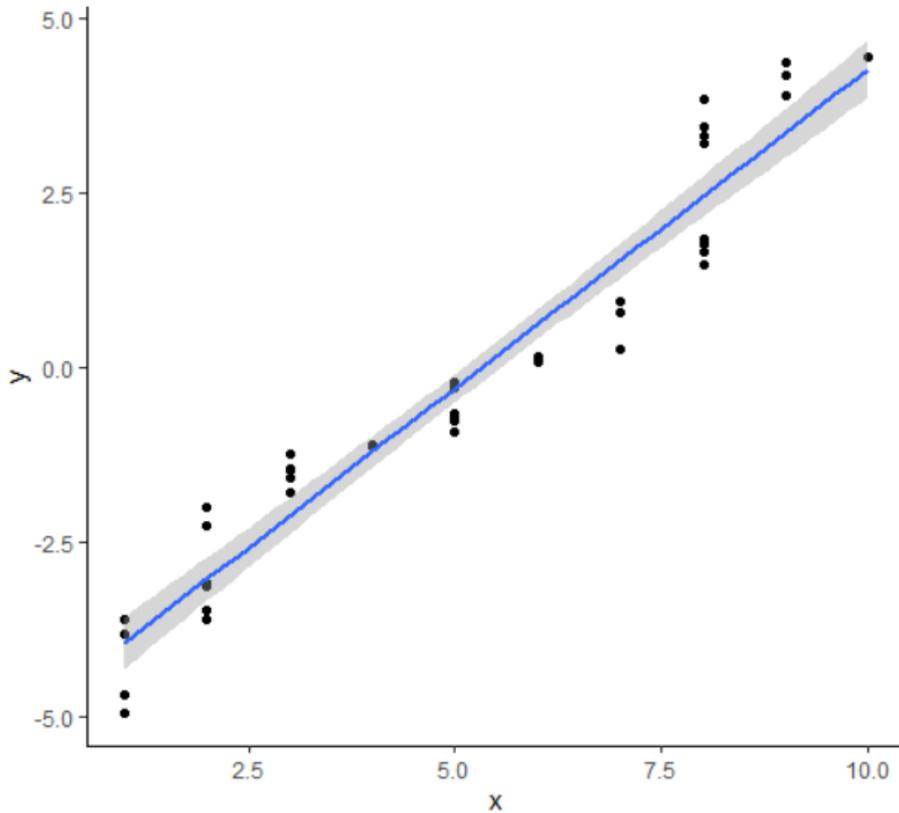
Check output

```
>coefficients (Spiti.large.hhn)
```

```
> coefficients(Spiti.large.hhn)
```

	beta	SE.beta	lcl	uc1
D	-9.509379	0.19995117	-9.901276	-9.117482
lambda0	1.043891	0.23207541	0.589032	1.498751
sigma	7.984838	0.08802027	7.812321	8.157354

Why exponential scale?



Check output

```
>predict (Spiti.large.hhn)
```

```
> predict(Spiti.large.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	7.415305e-05	1.497643e-05	5.011068e-05	1.097306e-04
lambda0	log	2.840248e+00	6.681275e-01	1.802243e+00	4.476095e+00
sigma	log	2.936100e+03	2.589377e+02	2.470859e+03	3.488943e+03

```
> coefficients(Spiti.large.hhn)
```

	beta	SE.beta	lcl	ucl
D	-9.509379	0.19995117	-9.901276	-9.117482
lambda0	1.043891	0.23207541	0.589032	1.498751
sigma	7.984838	0.08802027	7.812321	8.157354

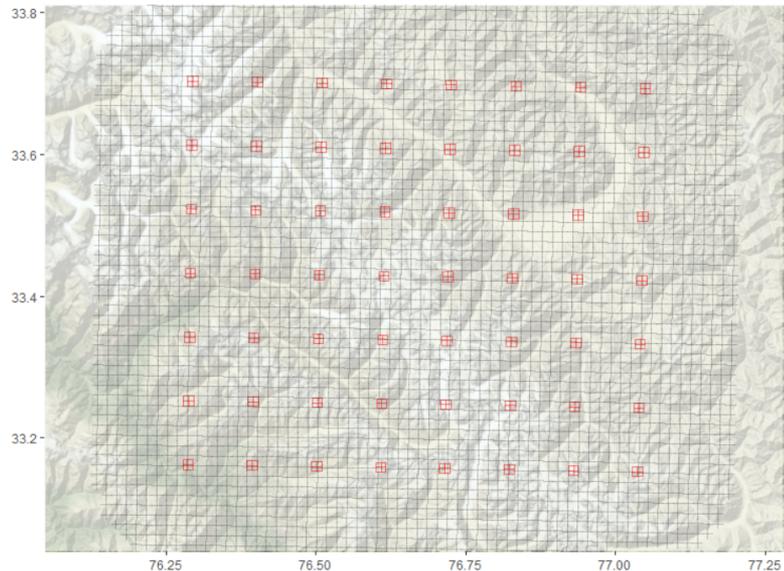
$$D = \exp(\beta_D)$$

Check output

```
region.N(Spiti.large.hhn)
```

```
> region.N(Spiti.large.hhn)
```

	estimate	SE.estimate	lcl	ucl	n
E.N	65.06496	13.14094	43.96918	96.28218	32
R.N	72.20705	10.37397	56.44609	98.12946	32

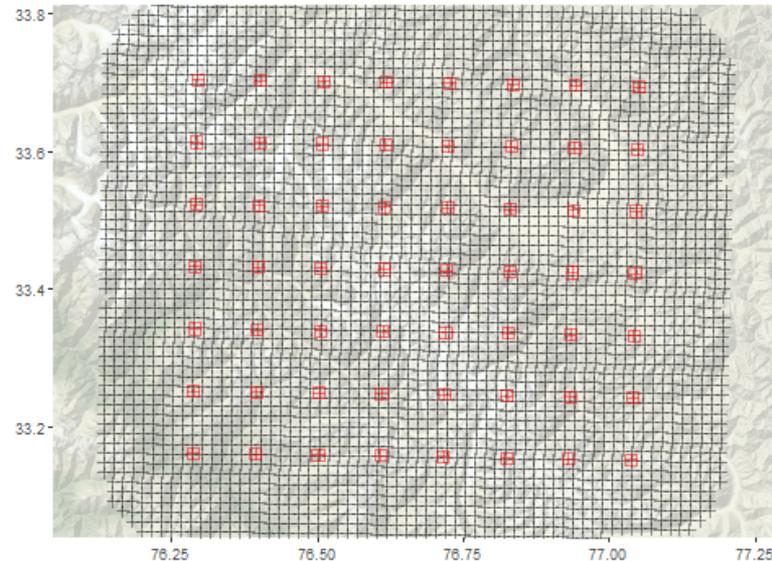


We use parameters to improve buffer

```
suggest.large.buff <-  
suggest.buffer(all.data.large,  
detectfn="HHN",  
detectpar=list(  
lambda0=predict(Spiti.large.hhn)[2,2],  
sigma=predict(Spiti.large.hhn)[3,2]),  
RBtarget = 0.001)
```

```
> suggest.large.buff
```

```
[1] 9561
```



Run a new model

```
Spiti.large.hhn2<-  
secr.fit(all.data.large,  
model =list(D~1, lambda0~1, sigma~1),  
detectfn="HHN",  
mask=Spiti.mask.large.2)
```

```
> predict(Spiti.large.hhn2)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	7.419392e-05	1.497048e-05	5.015656e-05	1.097511e-04
lambda0	log	2.838140e+00	6.684365e-01	1.799944e+00	4.475160e+00
sigma	log	2.936944e+03	2.595254e+02	2.470728e+03	3.491135e+03

```
> coefficients(Spiti.large.hhn2)
```

	beta	SE.beta	lcl	ucl
D	-9.508828	0.1997653	-9.9003612	-9.117295
lambda0	1.043149	0.2323478	0.5877553	1.498542
sigma	7.985125	0.0881940	7.8122679	8.157982

Compare the first and second

```
> predict(Spiti.large.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	7.415305e-05	1.497643e-05	5.011068e-05	1.097306e-04
lambda0	log	2.840248e+00	6.681275e-01	1.802243e+00	4.476095e+00
sigma	log	2.936100e+03	2.589377e+02	2.470859e+03	3.488943e+03

```
> predict(Spiti.large.hhn2)
```

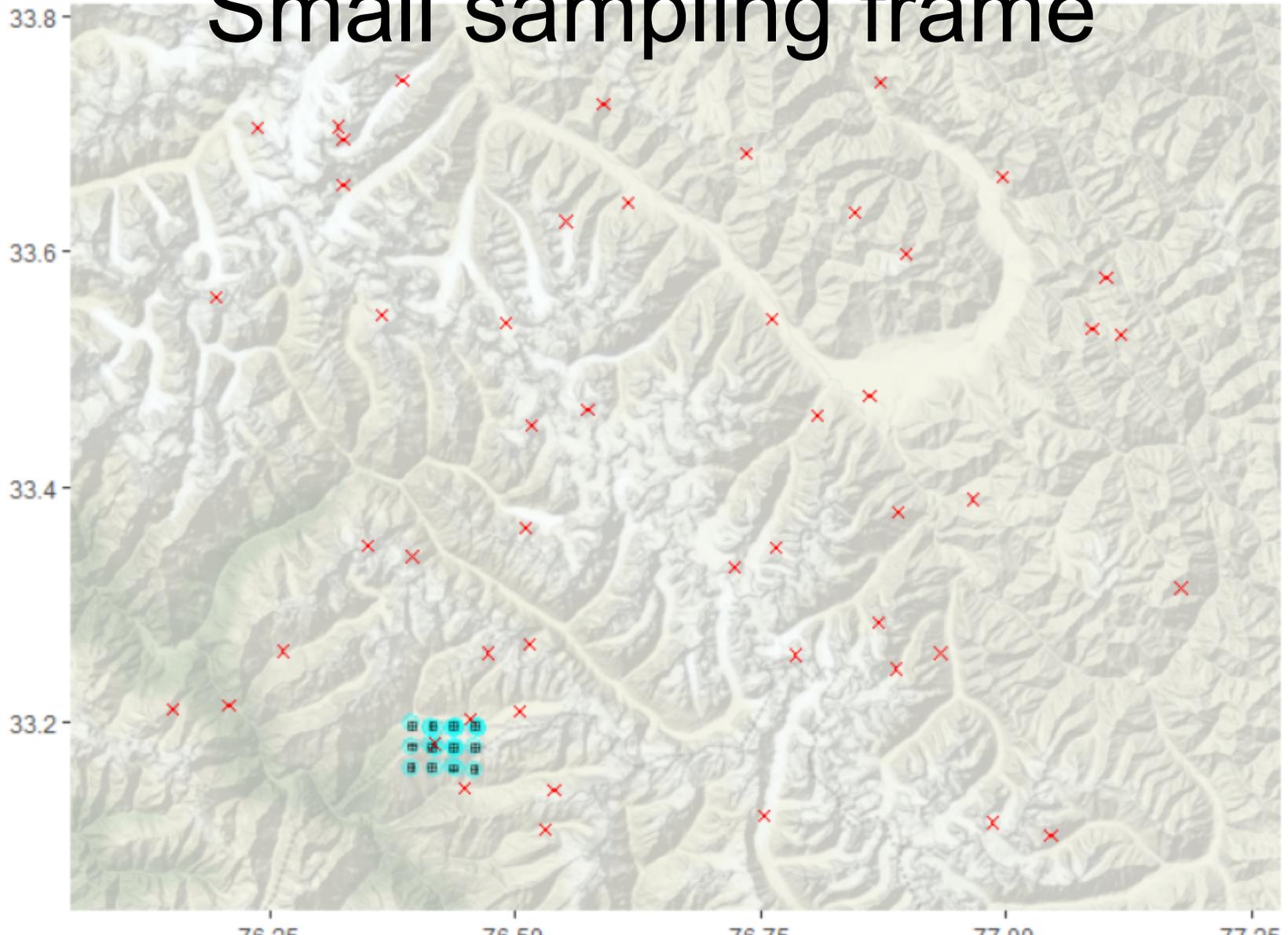
	link	estimate	SE.estimate	lcl	ucl
D	log	7.419392e-05	1.497048e-05	5.015656e-05	1.097511e-04
lambda0	log	2.838140e+00	6.684365e-01	1.799944e+00	4.475160e+00
sigma	log	2.936944e+03	2.595254e+02	2.470728e+03	3.491135e+03

Abundance WILL change!

```
> region.N(Spiti.large.hhn)
      estimate SE.estimate      lcl      ucl  n
E.N 65.06496   13.14094  43.96918  96.28218 32
R.N 72.20705   10.37397  56.44609  98.12946 32
```

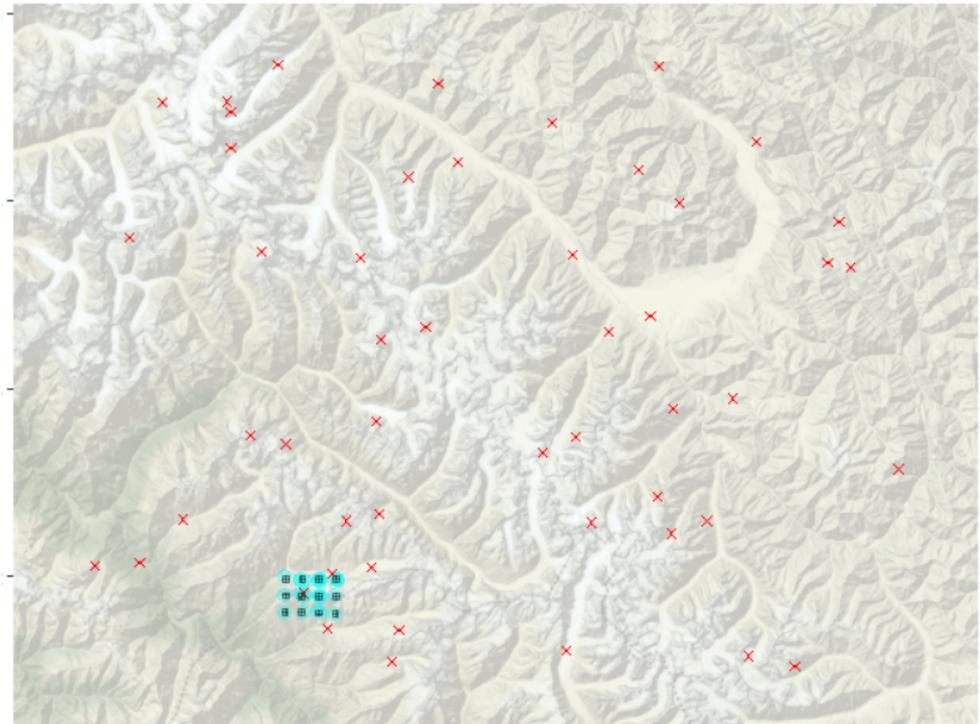
```
> region.N(Spiti.large.hhn2)
      estimate SE.estimate      lcl      ucl  n
E.N 52.08609   10.509669  35.21123  77.04817 32
R.N 59.23011    7.639834  47.87599  78.70441 32
```

Scenario 2: Small sampling frame



Capture data

	Session	ID	Occasion	trapID
1	1	7	1	C8
2	1	16	1	C2
3	1	16	1	C3
4	1	16	1	C12
5	1	16	1	C10
6	1	16	1	C3
7	1	16	1	C11
8	1	16	1	C8
9	1	16	1	C2
10	1	16	1	C10
11	1	16	1	C3
12	1	16	1	C4
13	1	16	1	C8



Trap data

	Session	ID	Occasion	trapID		x	y
1	1	7	1	C8	↔	C1	630000 3670000
2	1	16	1	C2	↔	C2	632000 3670000
3	1	16	1	C3	↔	C3	634000 3670000
4	1	16	1	C12	↔	C4	636000 3670000
5	1	16	1	C10	↔	C5	630000 3672000
6	1	16	1	C3	↔	C6	632000 3672000
7	1	16	1	C11	↔	C7	634000 3672000
8	1	16	1	C8	↔	C8	636000 3672000
9	1	16	1	C2	↔	C9	630000 3674000
10	1	16	1	C10	↔	C10	632000 3674000
11	1	16	1	C3	↔	C11	634000 3674000
12	1	16	1	C4	↔	C12	636000 3674000
13	1	16	1	C8	↔		

Output

```
> coefficients(Spiti.small.hhn)
```

	beta	SE.beta	lcl	ucl
D	-9.100868	0.6870950	-10.4475491	-7.754186
lambda0	1.026251	0.2563461	0.5238216	1.528680
sigma	8.255997	0.2982723	7.6713942	8.840600

```
> predict(Spiti.small.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	1.115690e-04	8.666237e-05	2.901931e-05	4.289431e-04
lambda0	log	2.790584e+00	7.272698e-01	1.688468e+00	4.612084e+00
sigma	log	3.850649e+03	1.174567e+03	2.146071e+03	6.909137e+03

Output

```
> predict(Spiti.small.hhn)
```

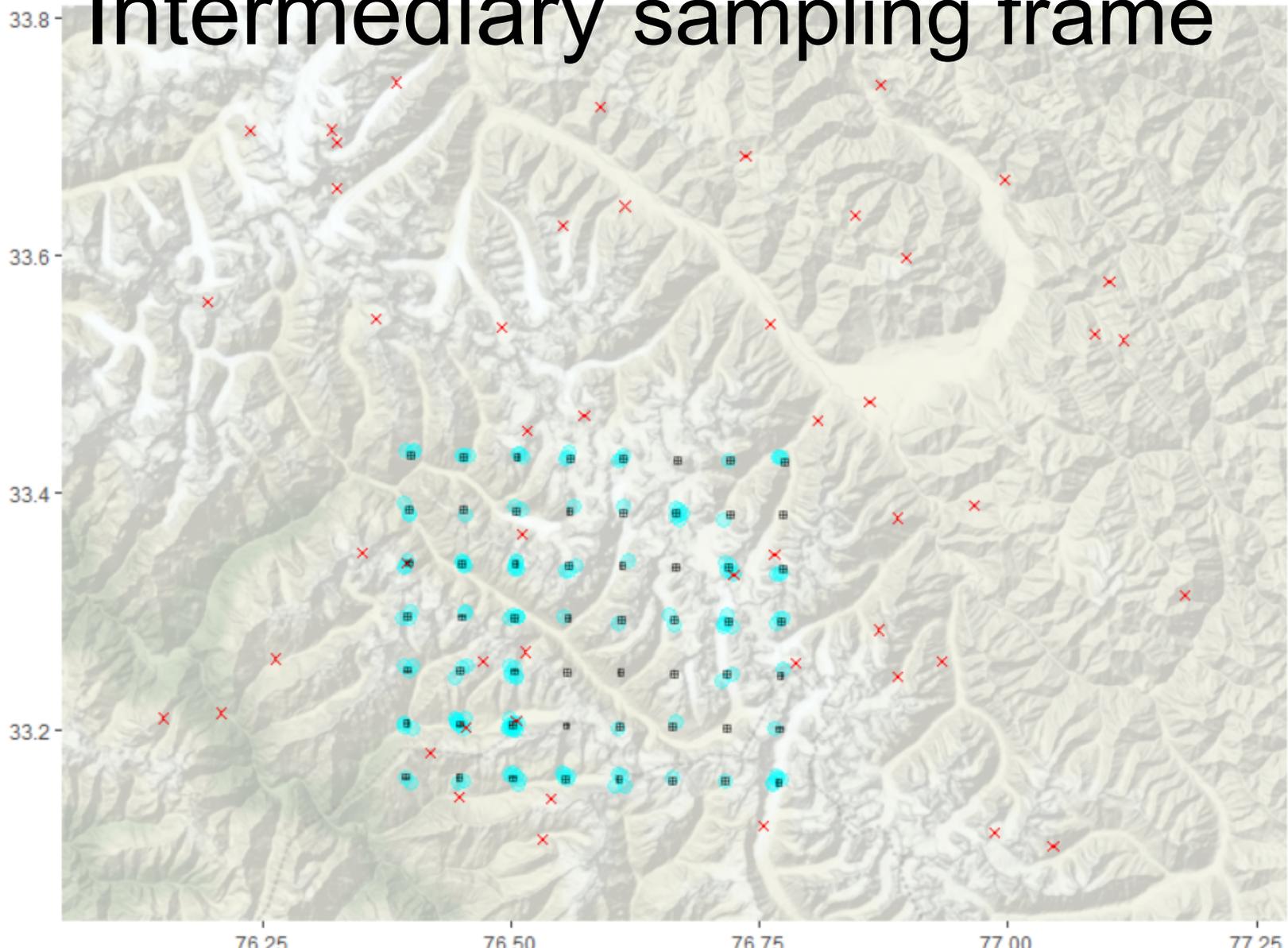
	link	estimate	SE.estimate	lcl	ucl
D	log	1.115690e-04	8.666237e-05	2.901931e-05	4.289431e-04
lambda0	log	2.790584e+00	7.272698e-01	1.688468e+00	4.612084e+00
sigma	log	3.850649e+03	1.174567e+03	2.146071e+03	6.909137e+03

```
> predict(Spiti.large.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	7.415305e-05	1.497643e-05	5.011068e-05	1.097306e-04
lambda0	log	2.840248e+00	6.681275e-01	1.802243e+00	4.476095e+00
sigma	log	2.936100e+03	2.589377e+02	2.470859e+03	3.488943e+03

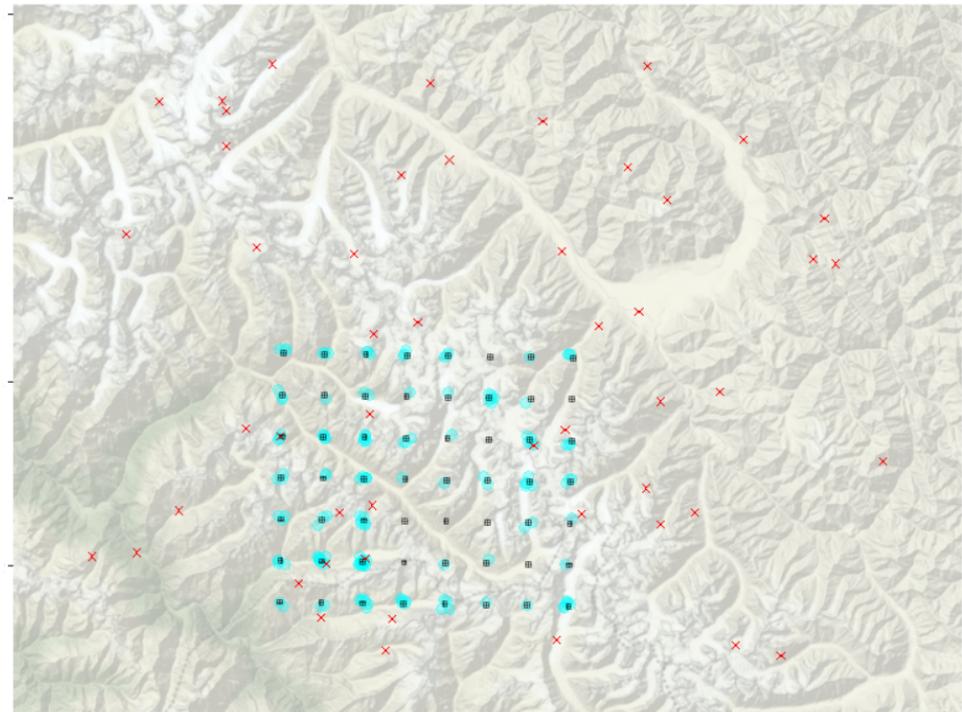
Scenario 3:

Intermediary sampling frame



Capture data

	Session	ID	Occasion	trapID
1	1	5	1	C24
2	1	7	1	C3
3	1	10	1	C45
4	1	10	1	C52
5	1	10	1	C44
6	1	10	1	C53
7	1	10	1	C52
8	1	10	1	C52
9	1	10	1	C43
10	1	10	1	C44
11	1	10	1	C52
12	1	10	1	C44
13	1	11	1	C31



Trap data

	Session	ID	Occasion	trapID
1	1	5	1	C24
2	1	7	1	C3
3	1	10	1	C45
4	1	10	1	C52
5	1	10	1	C44
6	1	10	1	C53
7	1	10	1	C52
8	1	10	1	C52
9	1	10	1	C43
10	1	10	1	C44
11	1	10	1	C52
12	1	10	1	C44
13	1	11	1	C31

	x	y
C1	630000	3675000
C2	635000	3675000
C3	640000	3675000
C4	645000	3675000
C5	650000	3675000
C6	655000	3675000
C7	660000	3675000
C8	665000	3675000
C9	630000	3680000
C10	635000	3680000
C11	640000	3680000
C12	645000	3680000
C13	650000	3680000



Output

```
> coefficients(Spiti.inter.hhn)
```

	beta	SE.beta	lcl	ucl
D	-9.395766	0.25207327	-9.8898209	-8.901712
lambda0	1.274644	0.16217376	0.9567895	1.592499
sigma	7.981694	0.06633159	7.8516865	8.111702

```
> predict(Spiti.inter.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	8.307503e-05	2.127809e-05	5.068802e-05	1.361556e-04
lambda0	log	3.577429e+00	5.840007e-01	2.603325e+00	4.916019e+00
sigma	log	2.926885e+03	1.943587e+02	2.570065e+03	3.333245e+03

Output

```
> predict(Spiti.inter.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	8.307503e-05	2.127809e-05	5.068802e-05	1.361556e-04
lambda0	log	3.577429e+00	5.840007e-01	2.603325e+00	4.916019e+00
sigma	log	2.926885e+03	1.943587e+02	2.570065e+03	3.333245e+03

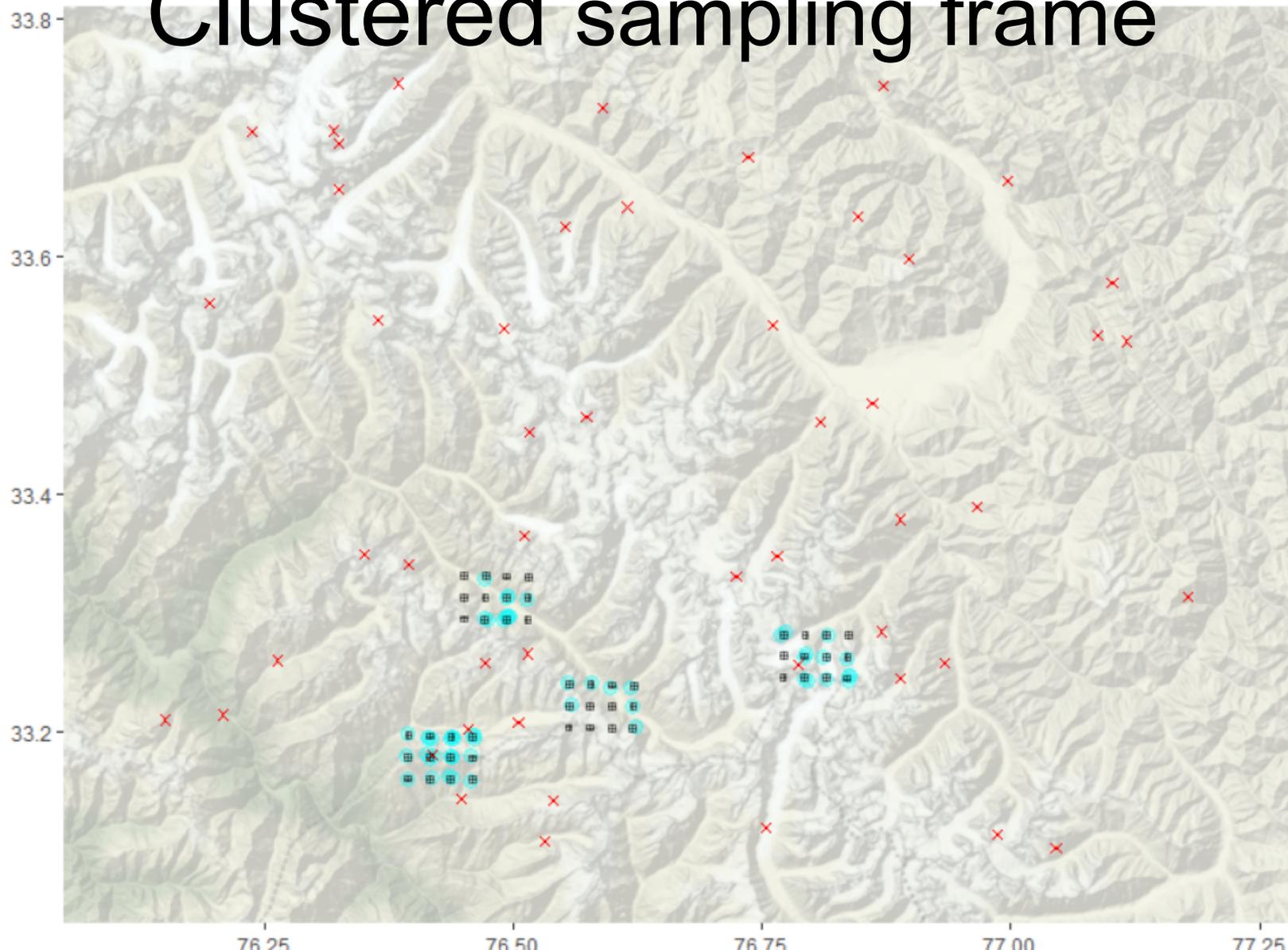
```
> predict(Spiti.small.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	1.115690e-04	8.666237e-05	2.901931e-05	4.289431e-04
lambda0	log	2.790584e+00	7.272698e-01	1.688468e+00	4.612084e+00
sigma	log	3.850649e+03	1.174567e+03	2.146071e+03	6.909137e+03

```
> predict(Spiti.large.hhn)
```

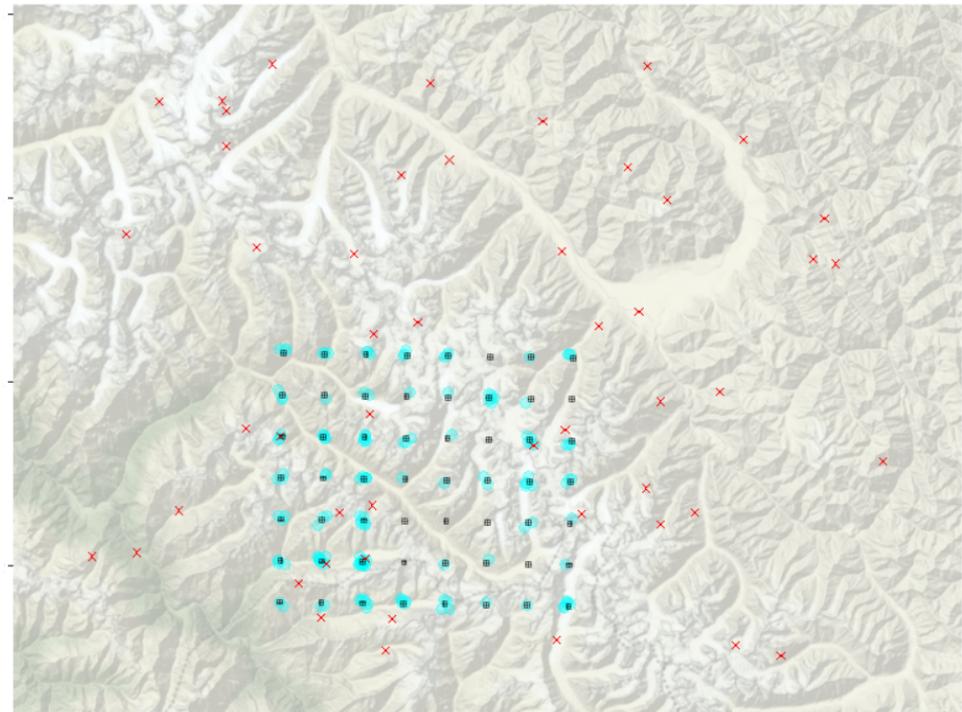
	link	estimate	SE.estimate	lcl	ucl
D	log	7.415305e-05	1.497643e-05	5.011068e-05	1.097306e-04
lambda0	log	2.840248e+00	6.681275e-01	1.802243e+00	4.476095e+00
sigma	log	2.936100e+03	2.589377e+02	2.470859e+03	3.488943e+03

Scenario 4: Clustered sampling frame



Capture data

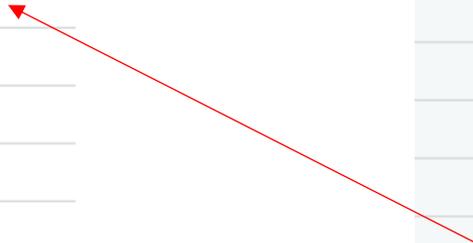
	Session	ID	Occasion	trapID
1	1	5	1	C36
2	1	5	1	C27
3	1	5	1	C27
4	1	7	1	C8
5	1	9	1	C31
6	1	9	1	C28
7	1	9	1	C32
8	1	11	1	C55
9	1	11	1	C33
10	1	12	1	C15
11	1	12	1	C50
12	1	12	1	C50
13	1	12	1	C16



Trap data

	Session	ID	Occasion	trapID
1	1	5	1	C36
2	1	5	1	C27
3	1	5	1	C27
4	1	7	1	C8
5	1	9	1	C31
6	1	9	1	C28
7	1	9	1	C32
8	1	11	1	C55
9	1	11	1	C33
10	1	12	1	C15
11	1	12	1	C50
12	1	12	1	C50
13	1	12	1	C16

	x	y
C1	630000	3670000
C2	632000	3670000
C3	634000	3670000
C4	636000	3670000
C5	630000	3672000
C6	632000	3672000
C7	634000	3672000
C8	636000	3672000
C9	630000	3674000
C10	632000	3674000
C11	634000	3674000
C12	636000	3674000
C13	635000	3685000



Output

```
> coefficients(Spiti.clust.hhn)
```

	beta	SE.beta	lcl	ucl
D	-9.228237	0.28211903	-9.781180	-8.675294
lambda0	1.025961	0.15042231	0.731139	1.320784
sigma	8.104982	0.05986228	7.987654	8.222310

```
> predict(Spiti.clust.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	9.822628e-05	2.827215e-05	5.650509e-05	1.707528e-04
lambda0	log	2.789776e+00	4.220296e-01	2.077446e+00	3.746356e+00
sigma	log	3.310921e+03	1.983770e+02	2.944381e+03	3.723091e+03

Output

```
> predict(Spiti.clust.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	9.822628e-05	2.827215e-05	5.650509e-05	1.707528e-04
lambda0	log	2.789776e+00	4.220296e-01	2.077446e+00	3.746356e+00
sigma	log	3.310921e+03	1.983770e+02	2.944381e+03	3.723091e+03

```
> predict(Spiti.inter.hhn)
```

	link	estimate	SE.estimate	lcl	ucl
D	log	8.307503e-05	2.127809e-05	5.068802e-05	1.361556e-04
lambda0	log	3.577429e+00	5.840007e-01	2.603325e+00	4.916019e+00
sigma	log	2.926885e+03	1.943587e+02	2.570065e+03	3.333245e+03

```
> predict(Spiti.small.hhn)
```

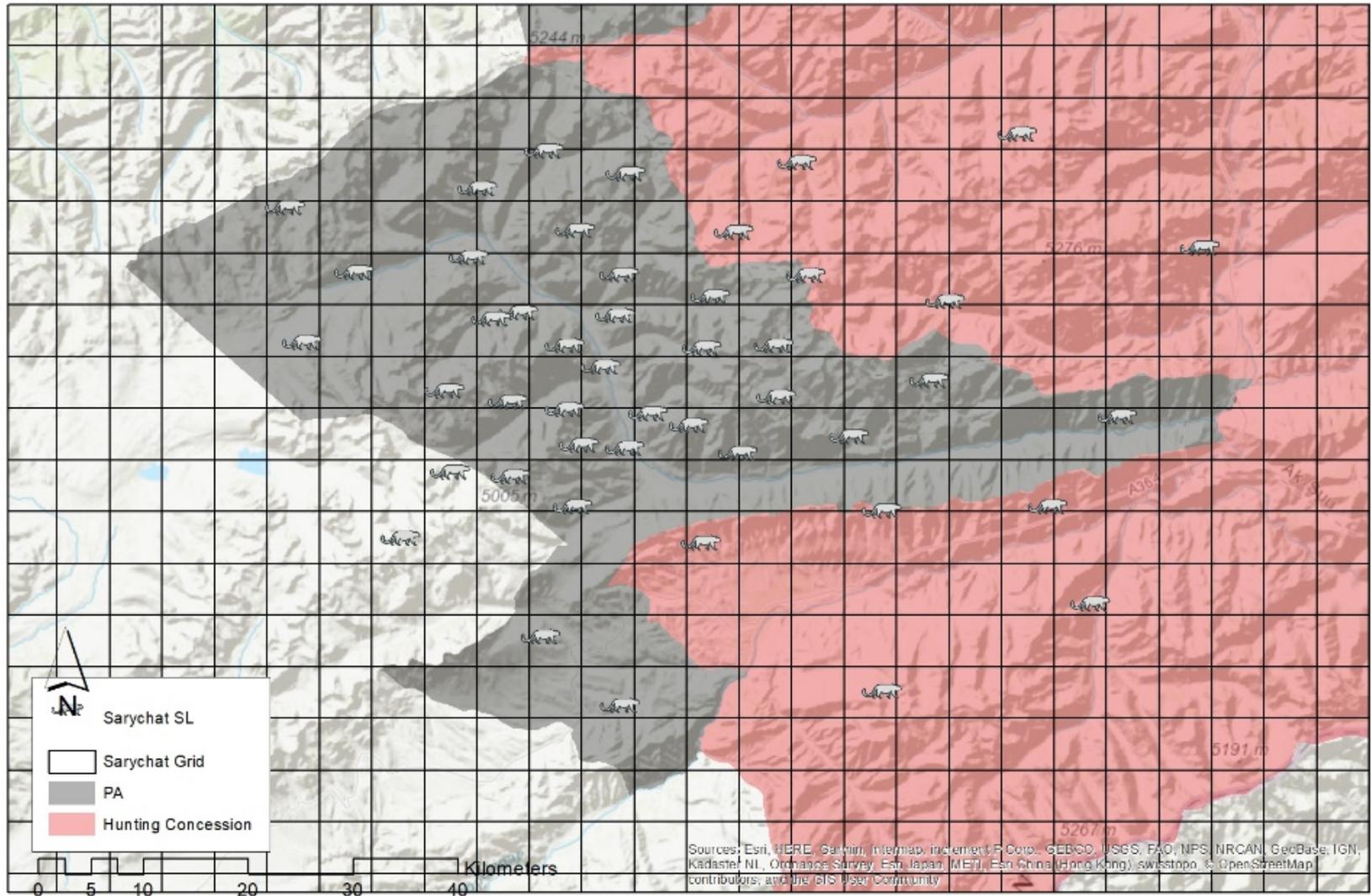
	link	estimate	SE.estimate	lcl	ucl
D	log	1.115690e-04	8.666237e-05	2.901931e-05	4.289431e-04
lambda0	log	2.790584e+00	7.272698e-01	1.688468e+00	4.612084e+00
sigma	log	3.850649e+03	1.174567e+03	2.146071e+03	6.909137e+03

```
> predict(Spiti.large.hhn)
```

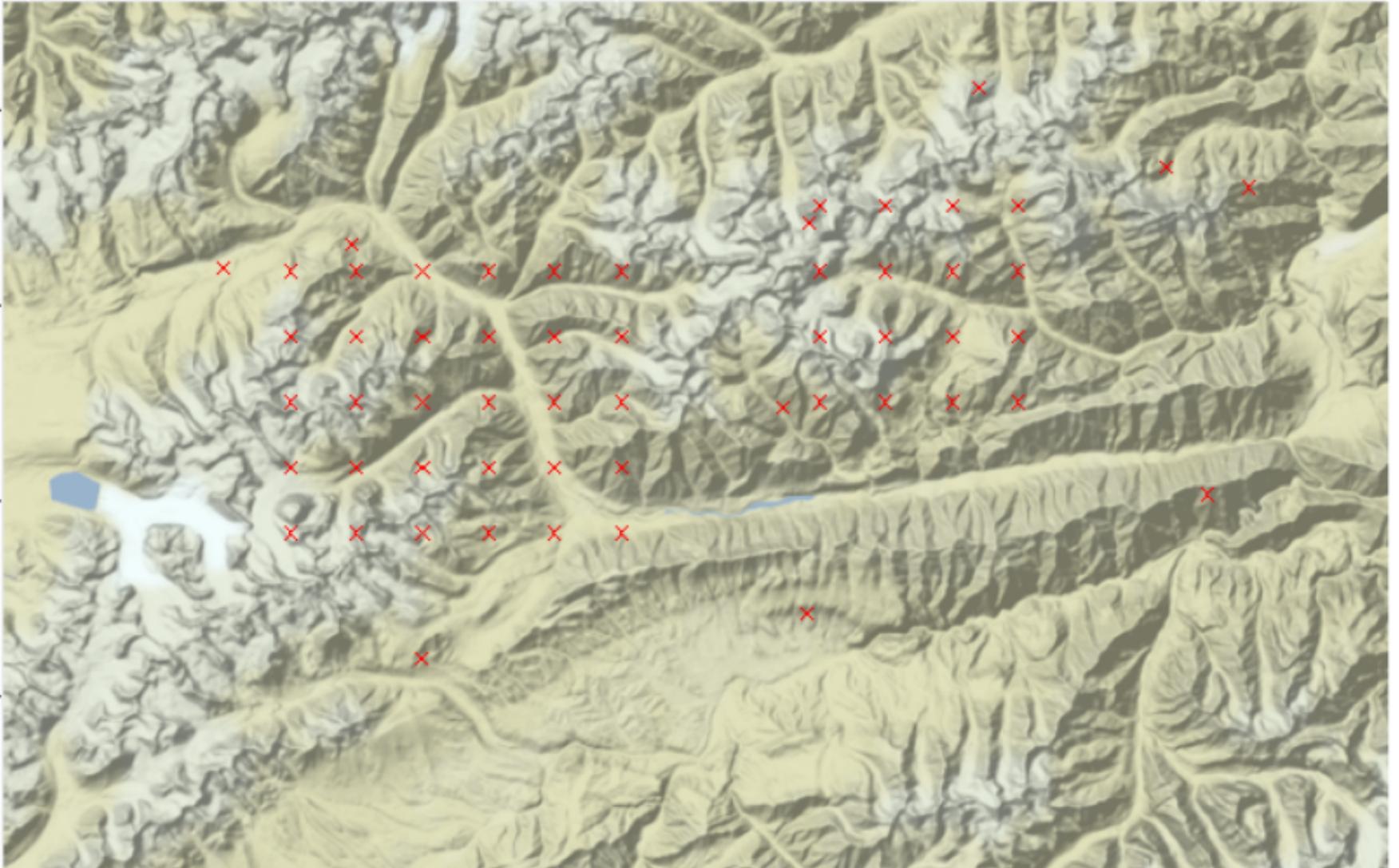
	link	estimate	SE.estimate	lcl	ucl
D	log	7.415305e-05	1.497643e-05	5.011068e-05	1.097306e-04
lambda0	log	2.840248e+00	6.681275e-01	1.802243e+00	4.476095e+00
sigma	log	2.936100e+03	2.589377e+02	2.470859e+03	3.488943e+03

Biases and Errors

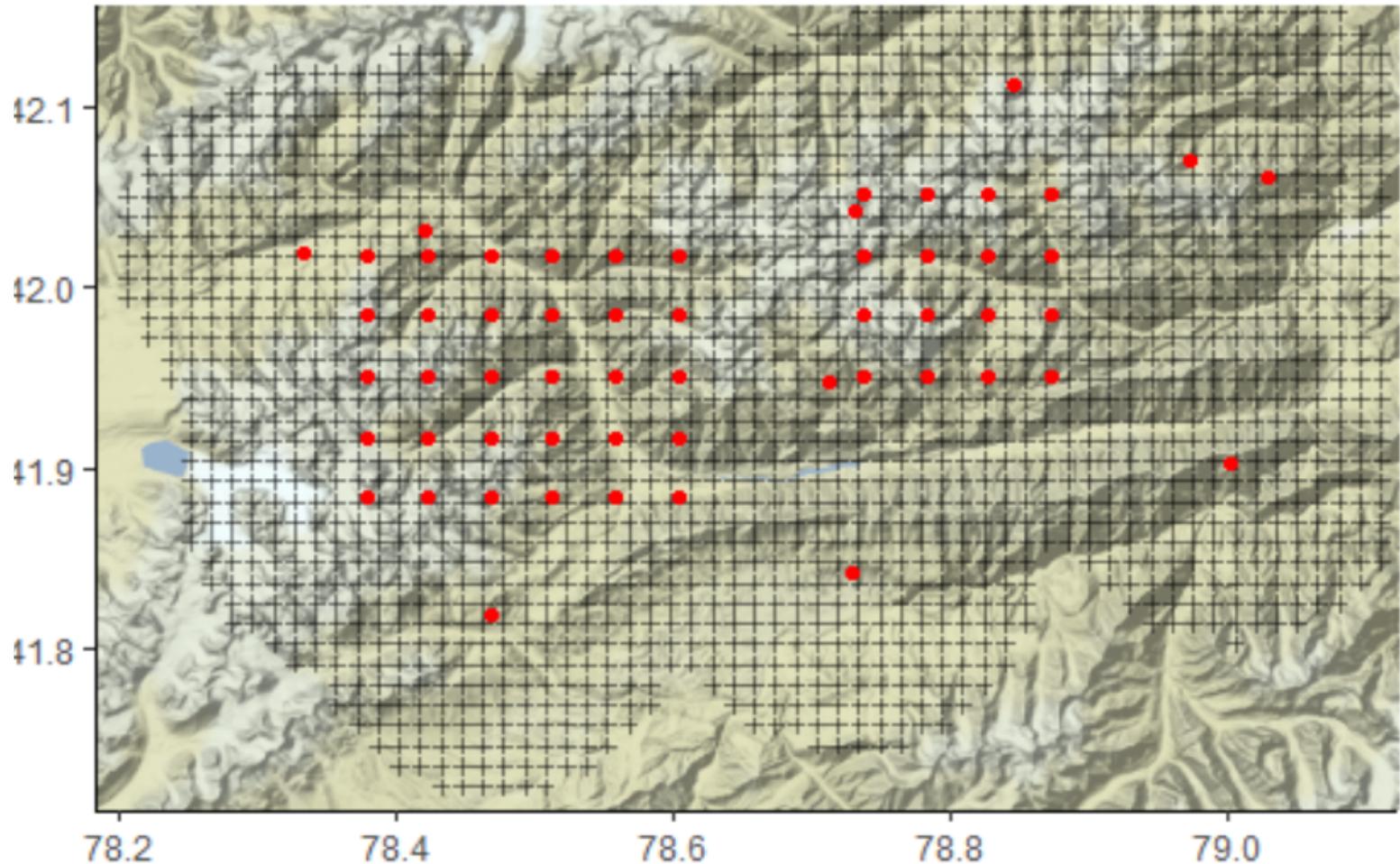
Non-uniform parameters



Setting 56 traps

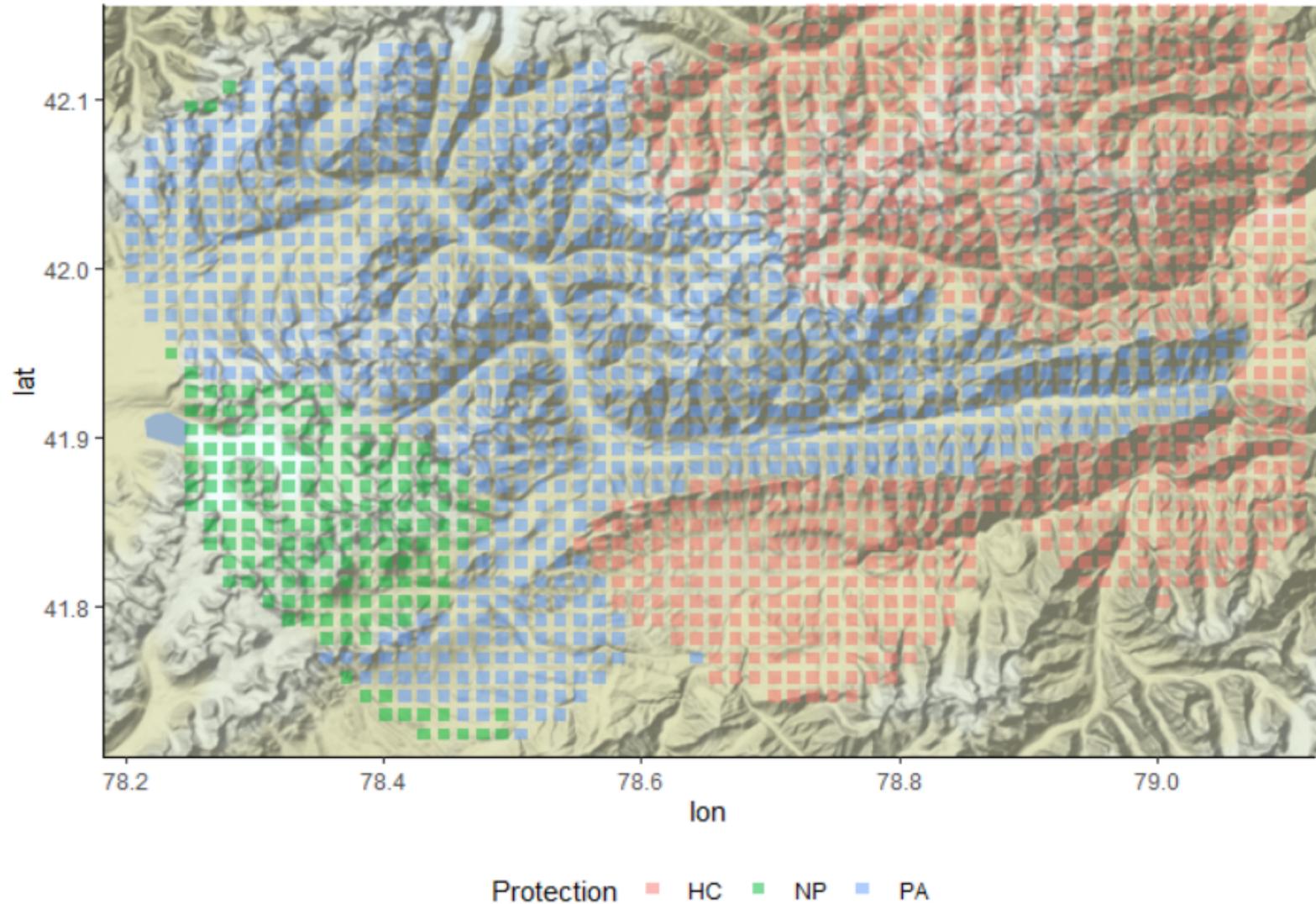


We create a mask!

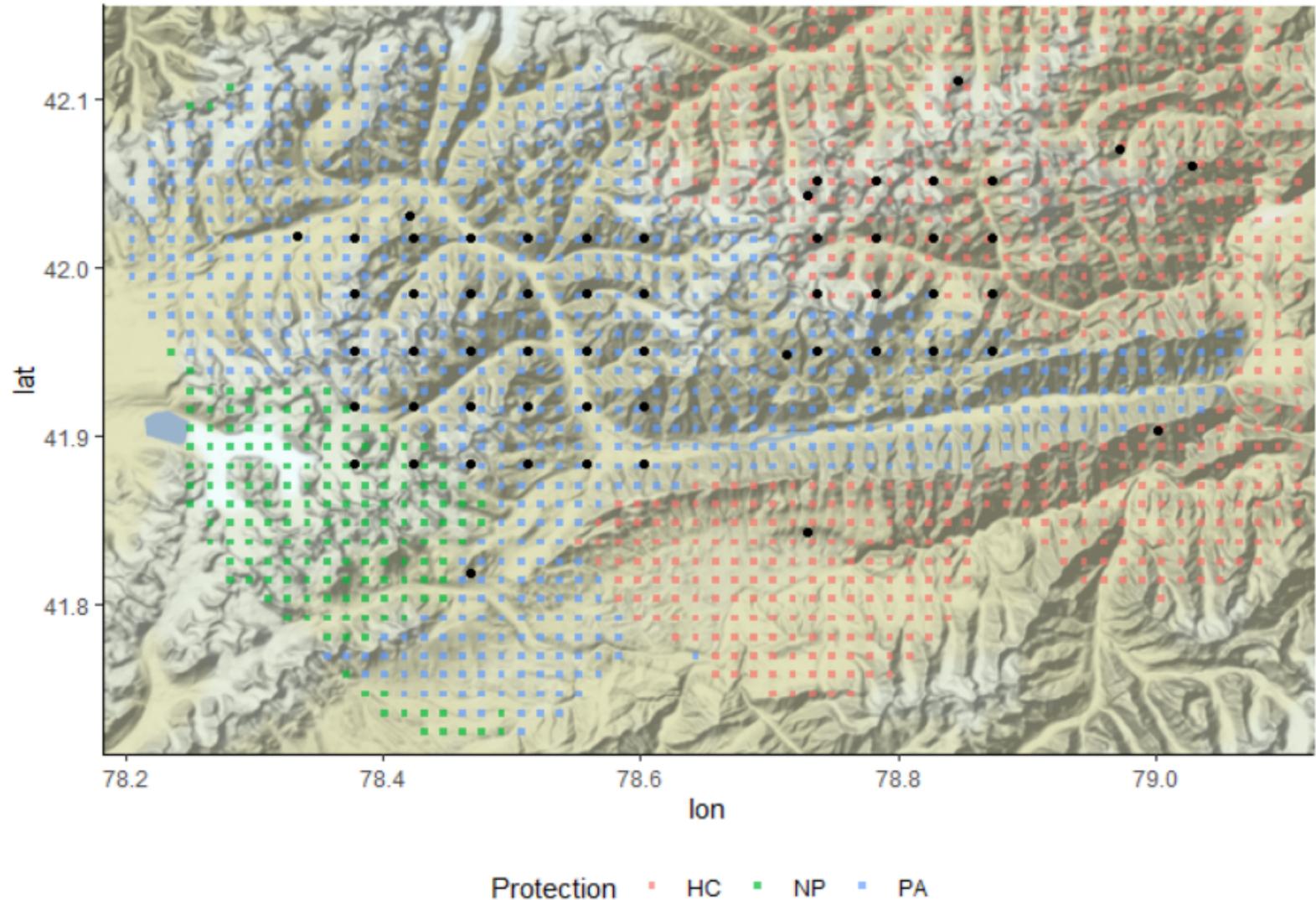


```
Sarychat.mask<-make.mask(traps(all.data.Sarychat),  
buffer = 15000, type = "trapbuffer")
```

Then we overlay the GIS data



and add covariates to mask



Check covariates

```
> head(covariates(SarychatMask.cov))  
> plot(SarychatMask.cov, covariate =  
"Protection", contour = FALSE, legend  
= FALSE)
```

	x	y	coords.x1	coords.x2	optional	FID	Protection	Dist_stl	std.Diststl
1	8731007	5091231	78.43197	41.72367	TRUE	0	NP	363.400	-0.66978008
2	8732685	5091231	78.44704	41.72367	TRUE	1	NP	310.838	-0.85446955
3	8734363	5091231	78.46211	41.72367	TRUE	2	NP	259.044	-1.03646054
4	8736040	5091231	78.47719	41.72367	TRUE	3	NP	225.457	-1.15447676
5	8737718	5091231	78.49226	41.72367	TRUE	4	NP	203.955	-1.23002932
6	8739396	5091231	78.50733	41.72367	TRUE	5	PA	181.172	-1.31008302
7	8727651	5092909	78.40182	41.73496	TRUE	6	NP	490.251	-0.22405781
8	8729329	5092909	78.41690	41.73496	TRUE	7	NP	435.532	-0.41632647
9	8731007	5092909	78.43197	41.73496	TRUE	8	NP	381.144	-0.60743210
10	8732685	5092909	78.44704	41.73496	TRUE	9	NP	327.702	-0.79521379
11	8734363	5092909	78.46211	41.73496	TRUE	10	PA	287.991	-0.93474816
12	8736040	5092909	78.47719	41.73496	TRUE	11	PA	264.444	-1.01748633

Modelling D ~ Protection

```
Spiti.large.hhn<-secr.fit(all.data.Sarychat,  
model= list(D~1, lambda0~1, sigma~1),  
detectfn="HHN",  
mask=SarychatMask.cov)
```

```
Sarychat.hhn.D_PA<-secr.fit(all.data.Sarychat,  
model = list(D~Protection, lambda0~1, sigma~1),  
detectfn="HHN", mask=SarychatMask.cov)
```

Outputs

```
> coefficients(Sarychat.hhn.D_PA)
```

	beta	SE.beta	lcl	uc1
D	-9.9604176	0.50463840	-10.9494906	-8.971344
D.ProtectionNP	0.8322921	0.93374714	-0.9978186	2.662403
D.ProtectionPA	1.3477827	0.54870709	0.2723365	2.423229
lambda0	1.0191855	0.11790522	0.7880956	1.250276
sigma	8.1253627	0.04752274	8.0322199	8.218506

Outputs

```
> coefficients(Sarychat.hhn)
```

	beta	SE.beta	lcl	ucl
D	-8.958818	0.17500611	-9.301824	-8.615813
lambda0	1.013881	0.11813664	0.782337	1.245424
sigma	8.120981	0.04751196	8.027859	8.214102

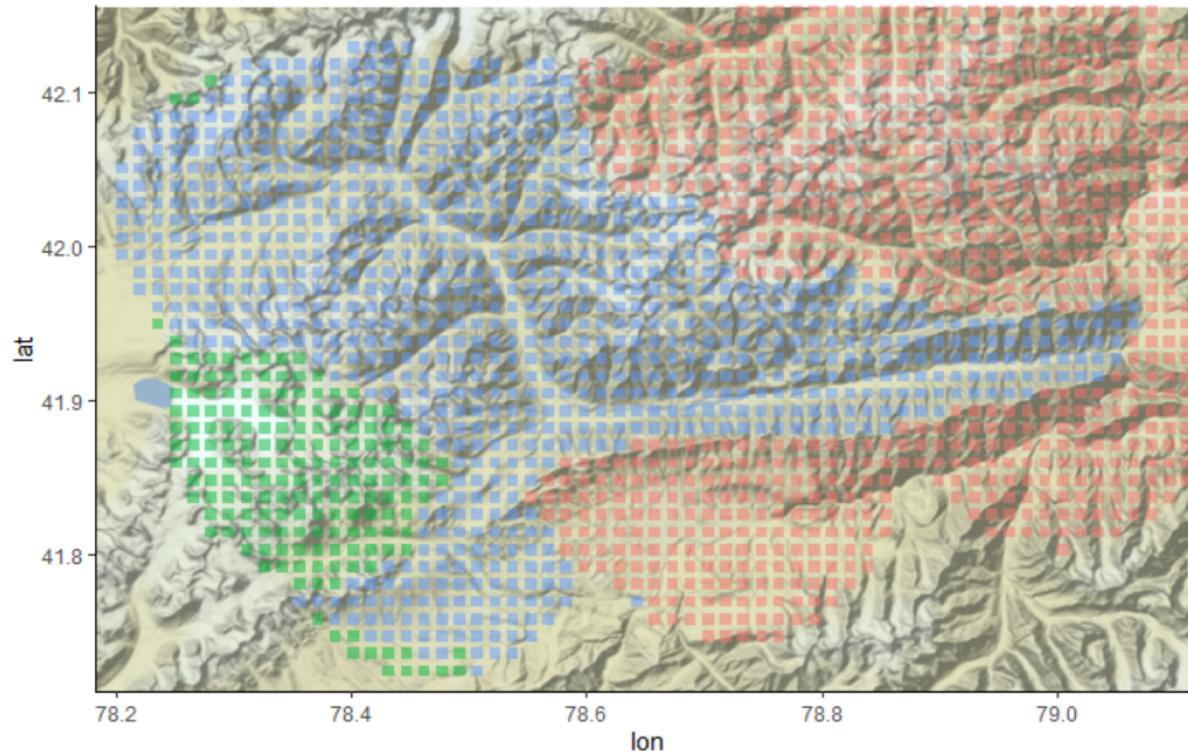
```
> coefficients(Sarychat.hhn.D_PA)
```

	beta	SE.beta	lcl	ucl
D	-9.9604176	0.50463840	-10.9494906	-8.971344
D.ProtectionNP	0.8322921	0.93374714	-0.9978186	2.662403
D.ProtectionPA	1.3477827	0.54870709	0.2723365	2.423229
lambda0	1.0191855	0.11790522	0.7880956	1.250276
sigma	8.1253627	0.04752274	8.0322199	8.218506

Predict doesn't work now!

```
> predict(Sarychat.hhn.D_PA)
```

	link	estimate	SE.estimate	lcl	uc1
D	log	4.723301e-05	2.543676e-05	1.756696e-05	1.269973e-04
lambda0	log	2.770937e+00	3.278467e-01	2.199204e+00	3.491305e+00
sigma	log	3.379093e+03	1.606745e+02	3.078568e+03	3.708956e+03



But overall density?

$$\begin{aligned} D &= \text{Abundance} / \text{Area} \\ &= \text{region.N}(\text{Sarychat.hhn.D_PA}) / \text{Mask Area} \\ &= 62.34579 / 5591 \\ &= 1.25 \quad (95\% \text{CI: } 1.02 - 1.59) \\ &\quad \text{per } 100 \text{ sq km} \end{aligned}$$

Which one to use?

AIC(Sarychat.hhn, Sarychat.hhn.D_PA)

```
> AIC(Sarychat.hhn, Sarychat.hhn.D_PA)
```

	model	detectfn	npar	logLik	AIC	AICc	dAICc	AICcwt
Sarychat.hhn.D_PA	D~Protection	lambda0~1 sigma~1 hazard halfnormal	5	-271.5130	553.026	555.169	0.000	0.8229
Sarychat.hhn	D~1	lambda0~1 sigma~1 hazard halfnormal	3	-275.7207	557.441	558.241	3.072	0.1771

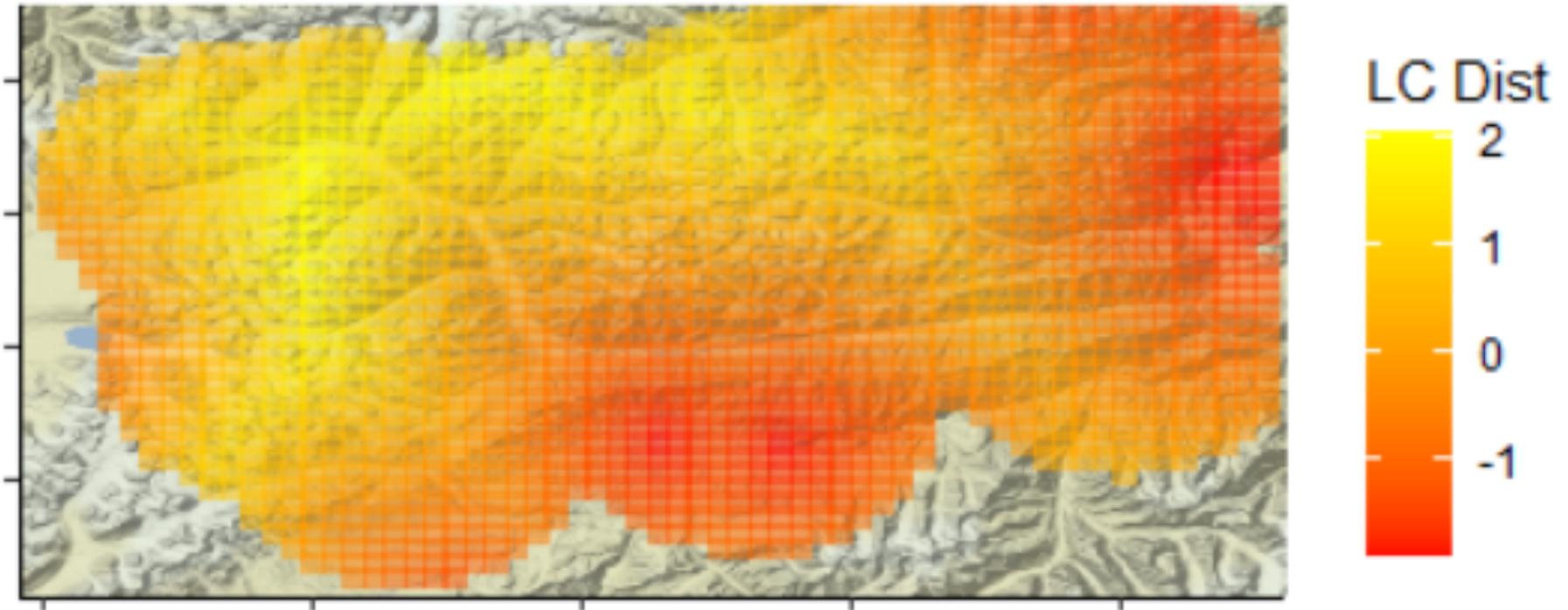
```
> region.N(Sarychat.hhn)
```

	estimate	SE.estimate	lcl	ucl	n
E.N	71.89936	12.679790	51.02224	101.3189	34
R.N	75.28396	9.427497	60.53800	98.2236	34

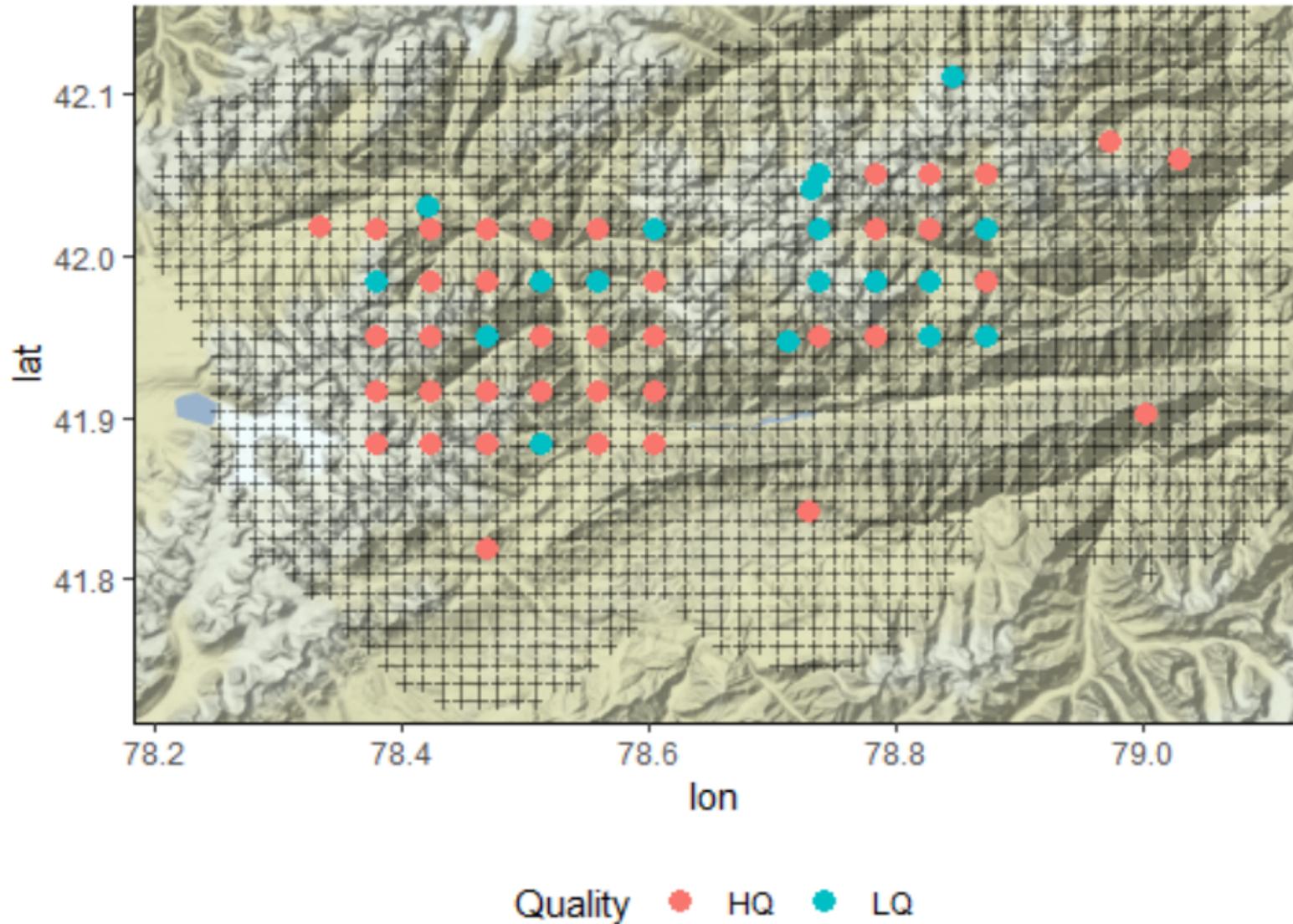
```
> region.N(Sarychat.hhn.D_PA)
```

	estimate	SE.estimate	lcl	ucl	n
E.N	62.34579	11.230434	43.92395	88.49381	34
R.N	69.63769	7.986042	57.09381	88.99503	34

We can test several relationships



Variable camera quality



Capture data

	Session	ID	Occasion	trapID
1	1	2	1	C41
2	1	2	1	C40
3	1	2	1	C40
4	1	2	1	C44
5	1	2	1	C41
6	1	2	1	C41
7	1	4	1	C49
8	1	5	1	C16
9	1	5	1	C16
10	1	5	1	C17
11	1	5	1	C16
12	1	5	1	C28
13	1	7	1	C16

Trap data

```
> table(Sarychat.captured.Q$Quality)
```

```
HQ  LQ  
132 7  
38 18
```

	Session	ID	Occasion	trapID
1	1	2	1	C41
2	1	2	1	C40
3	1	2	1	C40
4	1	2	1	C44
5	1	2	1	C41
6	1	2	1	C41
7	1	4	1	C49
8	1	5	1	C16
9	1	5	1	C16
10	1	5	1	C17
11	1	5	1	C16
12	1	5	1	C28
13	1	7	1	C16

	x	y	CamID	Quality
1	8725000	5115000	C1	HQ
2	8730000	5115000	C2	HQ
3	8735000	5115000	C3	HQ
4	8740000	5115000	C4	LQ
5	8745000	5115000	C5	HQ
6	8750000	5115000	C6	HQ
7	8725000	5120000	C7	HQ
8	8730000	5120000	C8	HQ
9	8735000	5120000	C9	HQ
10	8740000	5120000	C10	HQ
11	8745000	5120000	C11	HQ
12	8750000	5120000	C12	HQ
13	8725000	5125000	C13	HO

Modelling $\lambda_0 \sim$ Quality

```
Spiti.large.hhn<-secr.fit(all.data.Sarychat,  
model= list(D~1, lambda0~1, sigma~1),  
detectfn="HHN",  
mask=SarychatMask.cov)
```

```
Sarychat.hhn.D_PA<-secr.fit(all.data.Sarychat,  
model = list(D~Protection, lambda0~1, sigma~1),  
detectfn="HHN", mask=SarychatMask.cov)
```

```
Sarychat.hhn.l0_Q<-secr.fit(all.data.Sarychat.q,  
model = list(D~1, lambda0~Quality, sigma~1),  
detectfn="HHN", mask=SarychatMask.cov)
```

Output

```
> coefficients(Sarychat.hhn.10_Q)
```

	beta	SE.beta	lcl	uc1
D	-8.891911	0.17960240	-9.243925	-8.539897
lambda0	1.085988	0.13821278	0.815096	1.356880
lambda0.QualityLQ	-2.059148	0.40730102	-2.857443	-1.260852
sigma	8.116807	0.05389828	8.011168	8.222445

Modelling D ~ Protection

$\lambda_0 \sim$ Quality

```
Spiti.large.hhn<-secr.fit(all.data.Sarychat,  
model= list(D~1, lambda0~1, sigma~1),  
detectfn="HHN",  
mask=SarychatMask.cov)
```

```
Sarychat.hhn.D_PA<-secr.fit(all.data.Sarychat,  
model = list(D~Protection, lambda0~1, sigma~1),  
detectfn="HHN", mask=SarychatMask.cov)
```

```
Sarychat.hhn.l0_Q<-secr.fit(all.data.Sarychat.q, model =  
list(D~1, lambda0~Quality, sigma~1),  
detectfn="HHN", mask=SarychatMask.cov)
```

```
Sarychat.hhn.D_PA.l0_Q<-secr.fit(all.data.Sarychat.q,  
model = list(D~Protection, lambda0~Quality, sigma~1),  
detectfn="HHN", mask=SarychatMask.cov)
```

Output

```
> coefficients(Sarychat.hhn.10_Q)
```

	beta	SE.beta	lcl	uc1
D	-8.891911	0.17960240	-9.243925	-8.539897
lambda0	1.085988	0.13821278	0.815096	1.356880
lambda0.QualityLQ	-2.059148	0.40730102	-2.857443	-1.260852
sigma	8.116807	0.05389828	8.011168	8.222445

```
> coefficients(Sarychat.hhn.D_PA.10_Q)
```

	beta	SE.beta	lcl	uc1
D	-9.8045511	0.50085706	-10.7862129	-8.822889
D.ProtectionNP	0.6190232	0.94323543	-1.2296843	2.467731
D.ProtectionPA	1.2364823	0.54513686	0.1680337	2.304931
lambda0	1.0866412	0.13760222	0.8169458	1.356337
lambda0.QualityLQ	-2.0296356	0.40573677	-2.8248650	-1.234406
sigma	8.1235905	0.05393249	8.0178848	8.229296

Testing hypotheses!

> AIC(Sarychat.hhn.D_PA.10_Q, Sarychat.hhn.10_Q, Sarychat.hhn)

Model Name	model	npar	logLik	AIC	AICc	dAICc	AICcwt
Sarychat.hhn.D_PA.10_Q	D~Protection lambda0~Quality sigma~1	6	-220.356	452.712	455.943	0	0.6455
Sarychat.hhn.10_Q	D~1 lambda0~Quality sigma~1	4	-223.857	455.713	457.142	1.199	0.3545
Sarychat.hhn	D~1 lambda0~1 sigma~1	3	-275.721	557.441	558.241	102.298	0

Questions...

