

Estimating variance

David L Miller

Now we can make predictions

Now we are dangerous.

Predictions are useless without uncertainty

- We are doing statistics
- We want to know about **uncertainty**
- This is the most useful part of the analysis

What do we want the uncertainty for?

- Variance of total abundance
- Map of uncertainty (coefficient of variation)

Where does uncertainty come from?

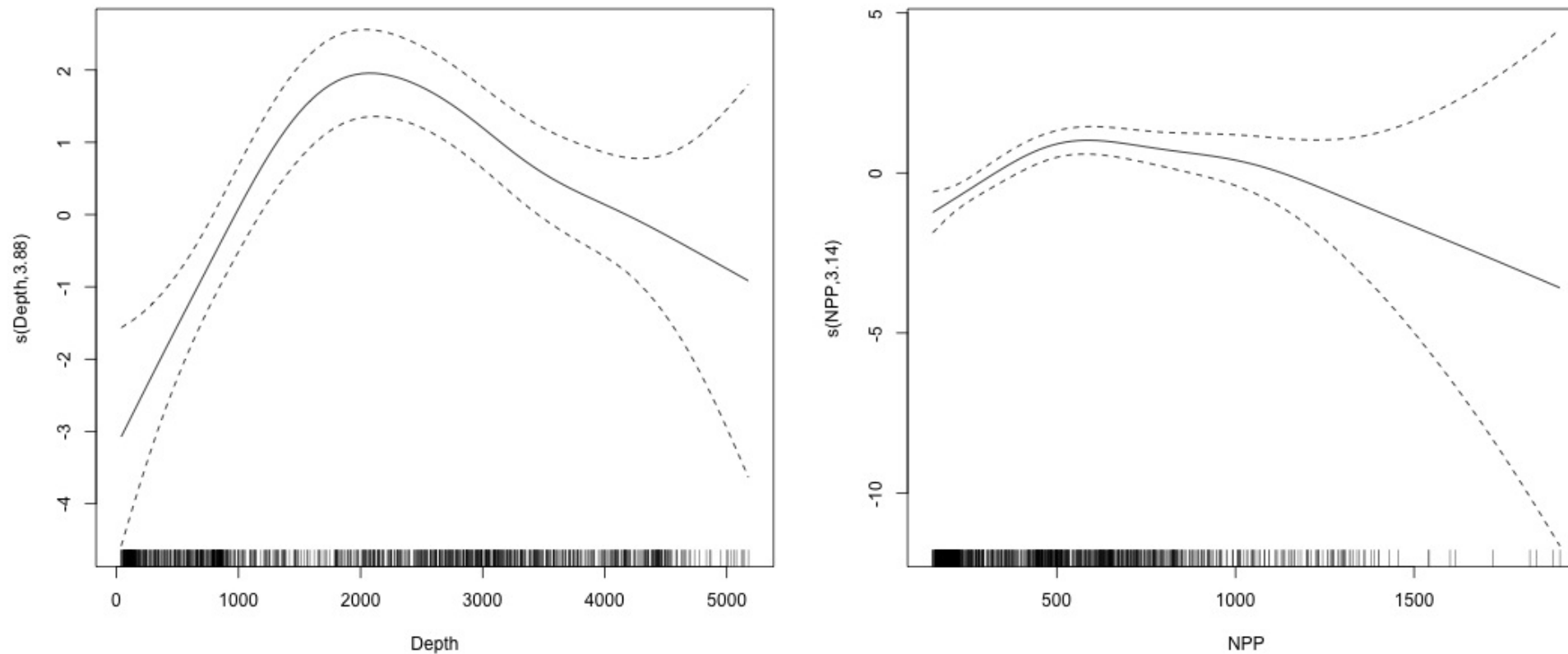
Sources of uncertainty

- Detection function
- GAM parameters

Let's think about smooths first

Uncertainty in smooths

- Dashed lines are ± 2 standard errors
- How do we translate to \hat{N} ?



Back to bases

- Before we expressed smooths as:
 - $s(x) = \sum_{k=1}^K \beta_k b_k(x)$
- Theory tells us that:
 - $\boldsymbol{\beta} \sim N(\hat{\boldsymbol{\beta}}, \mathbf{V}_{\boldsymbol{\beta}})$
 - where $\mathbf{V}_{\boldsymbol{\beta}}$ is a bit complicated
- Apply parameter variance to \hat{N}

Predictions to prediction variance (roughly)

- “map” data onto fitted values $\mathbf{X}\boldsymbol{\beta}$
- “map” prediction matrix to predictions $\mathbf{X}_p\boldsymbol{\beta}$
- Here \mathbf{X}_p need to take smooths into account
- pre-/post-multiply by \mathbf{X}_p to “transform variance”
 - $\Rightarrow \mathbf{X}_p^T \mathbf{V}_\beta \mathbf{X}_p$
 - link scale, need to do another transform for response

Adding in detection functions

GAM + detection function uncertainty

(Getting a little fast-and-loose with the mathematics)

From previous lectures we know:

$$\text{CV}^2(\hat{N}) \approx \text{CV}^2(\text{GAM}) + \text{CV}^2(\text{detection function})$$

Not that simple...

- Assumes detection function and GAM are **independent**
- Maybe this is okay?

A better way (for some models)

- Include the detectability as a “fixed” term in GAM
- Mean effect is zero
- Variance effect included
- Uncertainty “propagated” through the model
- Details in bibliography (too much to detail here)

That seemed complicated...

R to the rescue

In R...

- Functions in `dsm` to do this
- `dsm.var.gam`
 - assumes spatial model and detection function are independent
- `dsm.var.prop`
 - propagates uncertainty from detection function to spatial model
 - only works for count models (more or less)

Variance of abundance

Using `dsm.var.prop`

```
dsm_tw_var <- dsm.var.prop(dsm_all_tw_rm, predgrid,  
  off.set=predgrid$off.set)  
summary(dsm_tw_var)
```

Summary of uncertainty in a density surface model calculated
by variance propagation.

Quantiles of differences between fitted model and variance model

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-4.665e-04	-3.535e-05	-4.358e-06	-3.991e-06	2.095e-06	1.232e-03

Approximate asymptotic confidence interval:

5%	Mean	95%
1460.721	2491.914	4251.075

(Using delta method)

Point estimate	: 2491.914
Standard error	: 691.8776
Coefficient of variation	: 0.2776

Variance of abundance

Now using `dsm.var.gam`

```
dsm_tw_var_ind <- dsm.var.gam(dsm_all_tw_rm, predgrid,  
off.set=predgrid$off.set)  
summary(dsm_tw_var_ind)
```

Summary of uncertainty in a density surface model calculated
analytically for GAM, with delta method

Approximate asymptotic confidence interval:

5%	Mean	95%
1538.968	2491.864	4034.773

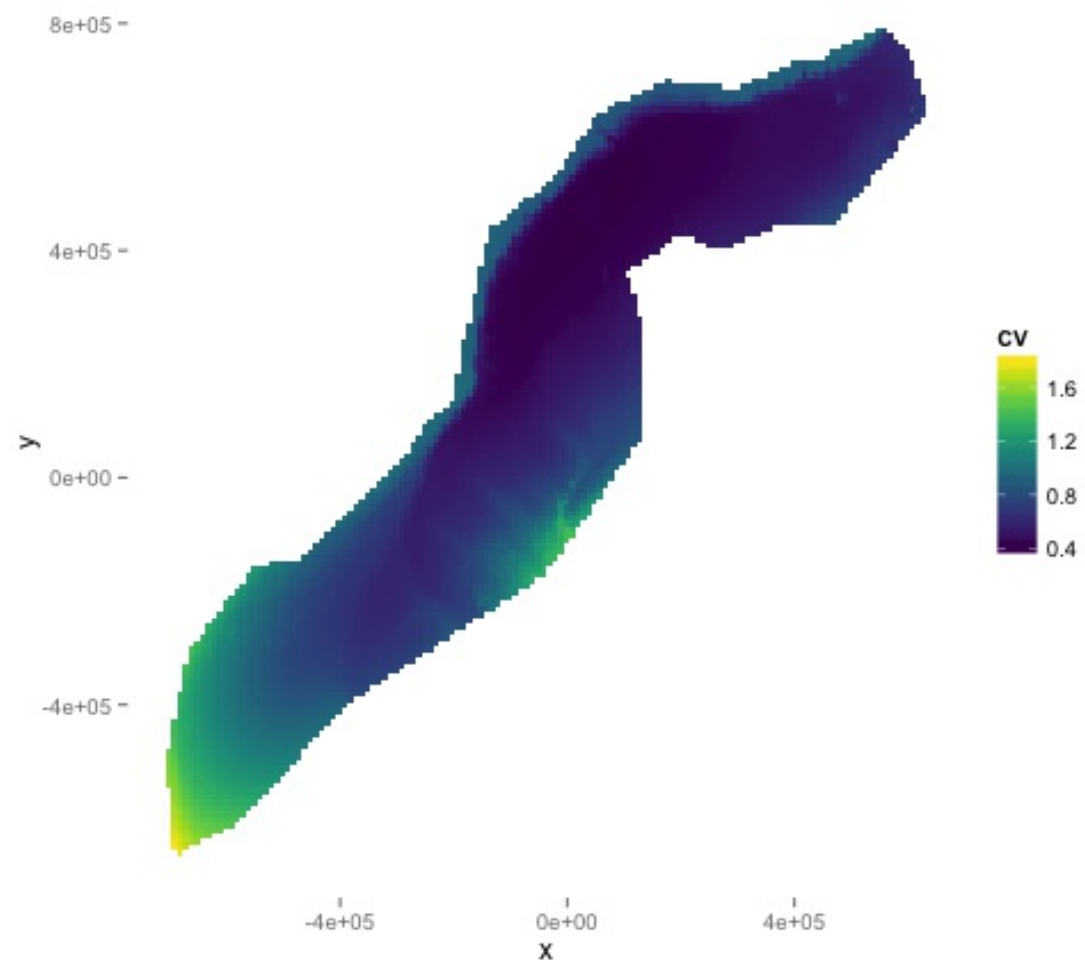
(Using delta method)

Point estimate	: 2491.864
Standard error	: 331.1575
CV of detection function	: 0.211327
CV from GAM	: 0.1329
Total coefficient of variation	: 0.2496

Plotting - data processing

- Calculate uncertainty per-cell
- `dsm.var.*` thinks `predgrid` is one “region”
- Need to split data into cells (using `split()`)
- (Could be arbitrary sets of cells, see exercises)
- Need `width` and `height` of cells for plotting

CV plot

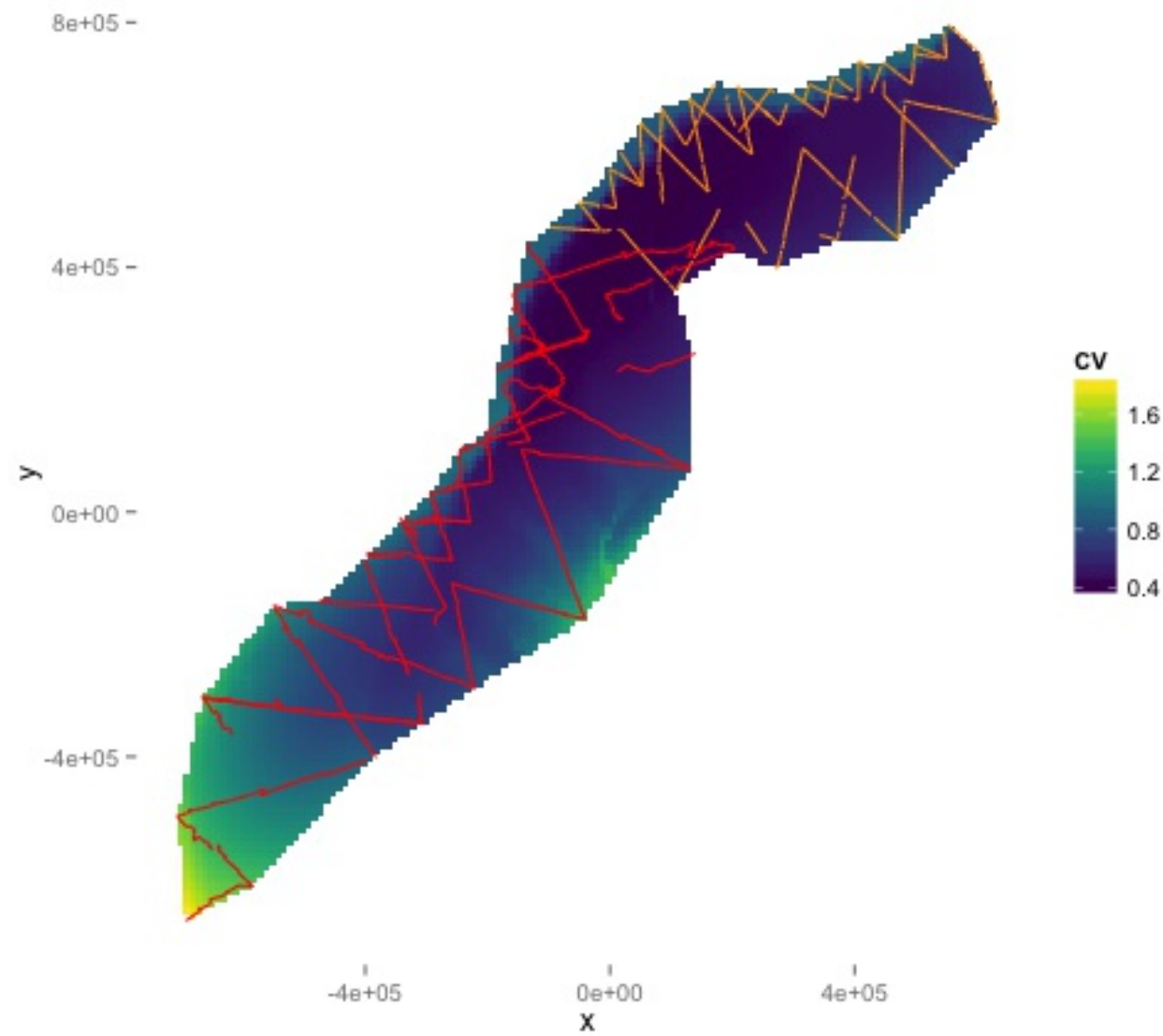


```
p <- plot(dsm_tw_var_map,
observations=FALSE, plot=FALSE)
+
+ coord_equal() +
+ scale_fill_viridis()
print(p)
```

Interpreting CV plots

- Plotting coefficient of variation
- Standardise standard deviation by mean
- $CV = se(\hat{N})/\hat{N}$ (per cell)
- Can be useful to overplot survey effort

Effort overplotted



Recap

- How does uncertainty arise in a DSM?
- Estimate variance of abundance estimate
- Map coefficient of variation

Let's try that!