Multivariate smoothing, model selection

Recap

- How GAMs work
- How to include detection info
- Simple spatial-only models
- How to check those models

Univariate models are fun, but…

Ecology is not univariate

- Many variables affect distribution
- Want to model the **right** ones
- Select between possible models
  - Smooth term selection
  - Response distribution
- Large literature on model selection

Models with multiple smooths

Adding smooths

- Already know that + is our friend
- Add everything then remove smooth terms?

```
dsm_all <- dsm(count~s(x, y) +
                     s(Depth) +
                     s(DistToCAS) +
                     s(SST) +
                     s(EKE) +
                     s(NPP),
               ddf.obj=df_hr,
               segment.data=segs, observation.data=obs,
               family=tw())
```

Now we have a huge model, what do we do?

Smooth term selection

- Classically, two main approaches
- Both have problems
- Usually use p-values

Stepwise selection - path dependence

All possible subsets - computationally expensive (fishing?)

p-values

- p-values can calculate
- Test for zero effect of a smooth
- They are **approximate** for GAMs (but useful)
- Reported in summary

p-values example

```
summary(dsm_all)
```

```
Family: Tweedie(p=1.25)
Link function: log

Formula:
count ~ s(x, y) + s(Depth) + s(DistToCAS) + s(SST) + s(EKE) +
    s(NPP) + offset(off.set)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.6369     0.2752     -75   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
              edf Ref.df     F p-value
s(x,y)      5.236  7.169 1.233  0.2928
s(Depth)    3.568  4.439 6.640 1.6e-05 ***
s(DistToCAS) 1.000  1.000 1.503  0.2205
s(SST)      5.927  6.987 2.067  0.0407 *
s(EKE)      1.763  2.225 2.577  0.0696 .
s(NPP)      2.393  3.068 0.855  0.4680
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Shrinkage or extra penalties

- Use penalty to remove terms during fitting
- Two methods
- Basis s(..., bs="ts") - thin plate splines with shrinkage
  - nullspace should be shrunk less than the wiggly part
- dsm(..., select=TRUE) - extra penalty
  - no assumption of how much to shrink the nullspace

Shrinkage example

```
dsm_ts_all <- dsm(count~s(x, y, bs="ts") +
                        s(Depth, bs="ts") +
                        s(DistToCAS, bs="ts") +
                        s(SST, bs="ts") +
                        s(EKE, bs="ts") +
                        s(NPP, bs="ts"),
                  ddf.obj=df_hr,
                  segment.data=segs, observation.data=obs,
                  family=tw())
```

Shrinkage example

```
summary(dsm_ts_all)
```

```
Family: Tweedie(p=1.277)
Link function: log

Formula:
count ~ s(x, y, bs = "ts") + s(Depth, bs = "ts") + s(DistToCAS,
    bs = "ts") + s(SST, bs = "ts") + s(EKE, bs = "ts") + s(NPP,
    bs = "ts") + offset(off.set)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -20.260      0.234  -86.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(x,y)      1.888e+00     29 0.705 3.56e-06 ***
s(Depth)    3.679e+00      9 4.811 2.15e-10 ***
s(DistToCAS) 9.339e-05     9 0.000   0.6797
s(SST)      3.827e-01      9 0.063   0.2160
s(EKE)      8.196e-01      9 0.499   0.0178 *
s(NPP)      3.570e-04      9 0.000   0.8359
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Extra penalty example

```
dsm_sel <- dsm(count~s(x, y) +
                  s(Depth) +
                  s(DistToCAS) +
                  s(SST) +
                  s(EKE) +
                  s(NPP),
               ddf.obj=df_hr,
               segment.data=segs, observation.data=obs,
               family=tw(), select=TRUE)
```

Extra penalty example

```
summary(dsm_sel)
```

```
Family: Tweedie(p=1.266)
Link function: log

Formula:
count ~ s(x, y) + s(Depth) + s(DistToCAS) + s(SST) + s(EKE) +
    s(NPP) + offset(off.set)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.4285     0.2454  -83.23   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(x,y)      7.694e+00     29 1.272 2.67e-07 ***
s(Depth)    3.645e+00      9 4.005 3.24e-10 ***
s(DistToCAS) 1.944e-05      9 0.000   0.7038
s(SST)      2.010e-04      9 0.000   0.8216
s(EKE)      1.417e+00      9 0.630   0.0127 *
s(NPP)      2.318e-04      9 0.000   0.5152
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EDF comparison

	allterms	select	ts
s(x,y)	5.236	7.6936	1.8875
s(Depth)	3.568	3.6449	3.6794
s(DistToCAS)	1.000	0.0000	0.0001
s(SST)	5.927	0.0002	0.3827
s(EKE)	1.763	1.4174	0.8196
s(NPP)	2.393	0.0002	0.0004

Double penalty can be slow

- Lots of smoothing parameters to estimate

```
length(dsm_ts_all$sp)
```

```
[1] 6
```

```
length(dsm_sel$sp)
```

```
[1] 12
```

Let’s employ a mixture of these techniques

How do we select smooth terms?

1. Look at EDF
   - Terms with EDF<1 may not be useful
   - These can usually be removed
2. Remove non-significant terms by p-value
   - Decide on a significance level and use that as a rule

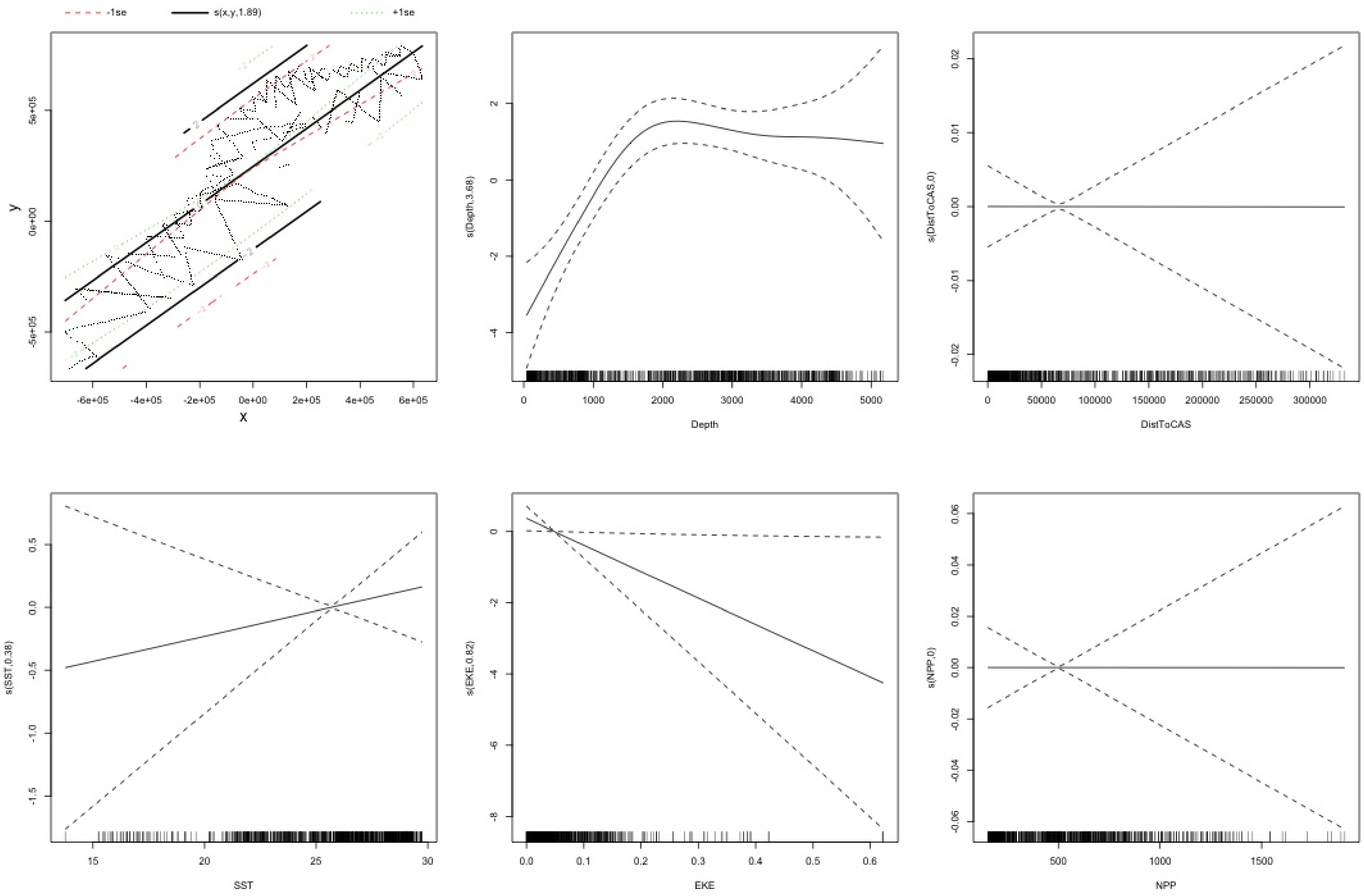
(In some sense leaving "shrunk" terms in is more "consistent", but can be computationally annoying)

Example of selection

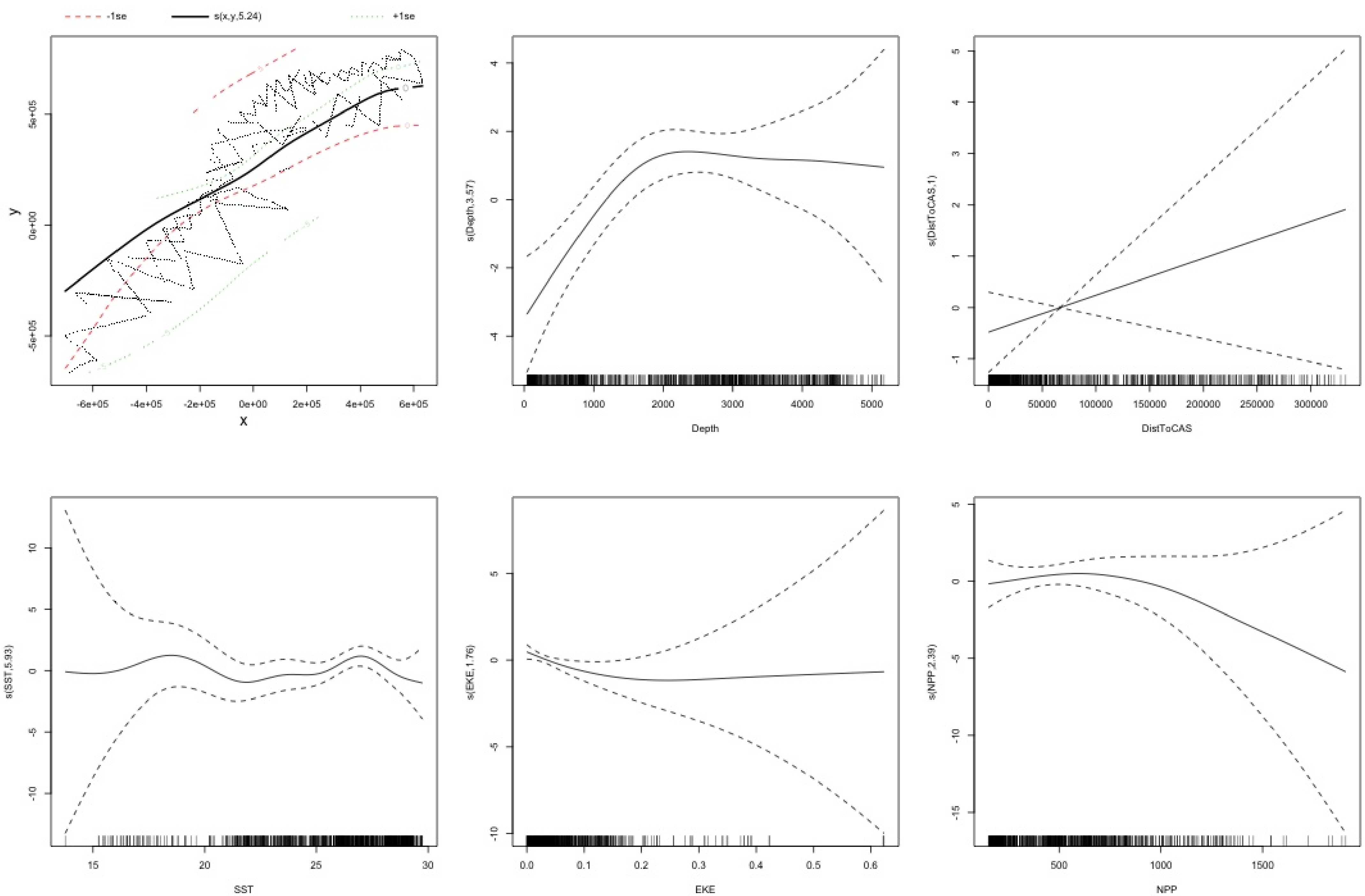
Selecting smooth terms

```
Family: Tweedie(p=1.277)
Link function: log

Formula:
count ~ s(x, y, bs = "ts") + s(Depth, bs = "ts") + s(DistToCAS,
    bs = "ts") + s(SST, bs = "ts") + s(EKE, bs = "ts") + s(NPP,
    bs = "ts") + offset(off.set)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -20.260      0.234  -86.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(x,y)      1.888e+00     29 0.705 3.56e-06 ***
s(Depth)    3.679e+00      9 4.811 2.15e-10 ***
s(DistToCAS) 9.339e-05     9 0.000   0.6797
s(SST)      3.827e-01      9 0.063   0.2160
s(EKE)      8.196e-01      9 0.499   0.0178 *
s(NPP)      3.570e-04      9 0.000   0.8359
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =   0.11   Deviance explained =   35%
-REML = 385.04  Scale est. = 4.5486    n = 949
```

Shrinkage in action



Same model with no shrinkage



Let's remove some smooth terms & refit

```
dsm_all_tw_rm <- dsm(count~s(x, y, bs="ts") +
                        s(Depth, bs="ts") +
                        #s(DistToCAS, bs="ts") +
                        #s(SST, bs="ts") +
                        s(EKE, bs="ts"),#+
                        #s(NPP, bs="ts"),
                     ddf.obj=df_hr,
                     segment.data=segs,
                     observation.data=obs,
                     family=tw())
```

What does that look like?

```
Family: Tweedie(p=1.279)
Link function: log

Formula:
count ~ s(x, y, bs = "ts") + s(Depth, bs = "ts") + s(EKE, bs = "ts") +
    offset(off.set)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -20.258      0.234  -86.56   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
            edf Ref.df     F  p-value
s(x,y)   1.8969     29 0.707 1.76e-05 ***
s(Depth) 3.6949      9 5.024 1.08e-10 ***
s(EKE)   0.8106      9 0.470   0.0216 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.105   Deviance explained = 34.8%
-REML = 385.09  Scale est. = 4.5733    n = 949
```

Removing EKE…

```
Family: Tweedie(p=1.268)
Link function: log

Formula:
count ~ s(x, y, bs = "ts") + s(Depth, bs = "ts") + offset(off.set)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -20.3088     0.2425  -83.75   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(x,y)   6.443     29 1.322 4.75e-08 ***
s(Depth) 3.611      9 4.261 1.49e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.141   Deviance explained = 37.8%
-REML = 389.86  Scale est. = 4.3516    n = 949
```

General strategy

For each response distribution and non-nested model structure:

1. Build a model with the smooths you want
2. Make sure that smooths are flexible enough (k=…)
3. Remove smooths that have been shrunk
4. Remove non-significant smooths

Comparing models

Comparing models

- Usually have >1 option
- How can we pick?
- Even if we have 1 model, is it any good?

Nested vs. non-nested models

- Compare ~s(x)+s(depth) with ~s(x)
  - nested models
- What about s(x) + s(y) vs. s(x, y)
  - don't want to have all these in the model
  - not nested models

Measures of ”fit”

- Two listed in summary
  - Deviance explained
  - Adjusted R^2
- Deviance is a generalisation of R^2
- Highest likelihood value (*saturated* model) minus estimated model value
- (These are usually not very high for DSMs)

AIC

- Can get AIC from our model
- Comparison of AIC fine (but not the end of the story)

```
AIC(dsm_all)
```

```
[1] 1238.307
```

```
AIC(dsm_ts_all)
```

```
[1] 1225.822
```

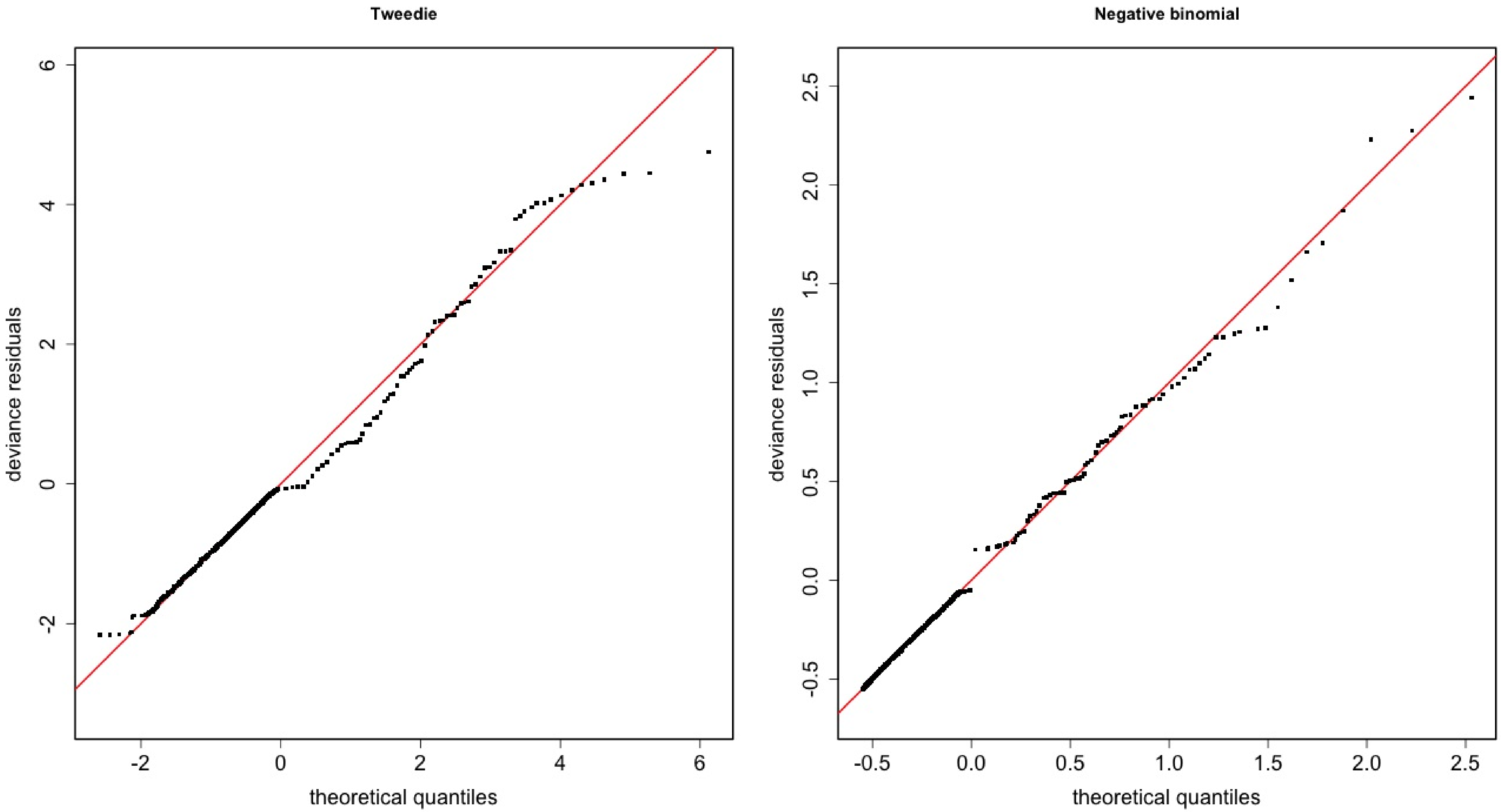
A quick note about REML scores

- Use REML to select the smoothness
- Can also use the score to do model selection
- **BUT** only compare models with the same fixed effects
  - (i.e., same “linear terms” in the model)
- $\Rightarrow$ **All terms** must be penalised
  - `bs="ts"` or `select=TRUE`

Selecting between response distributions

Goodness of fit tests

- Q-Q plots
- Closer to the line == better

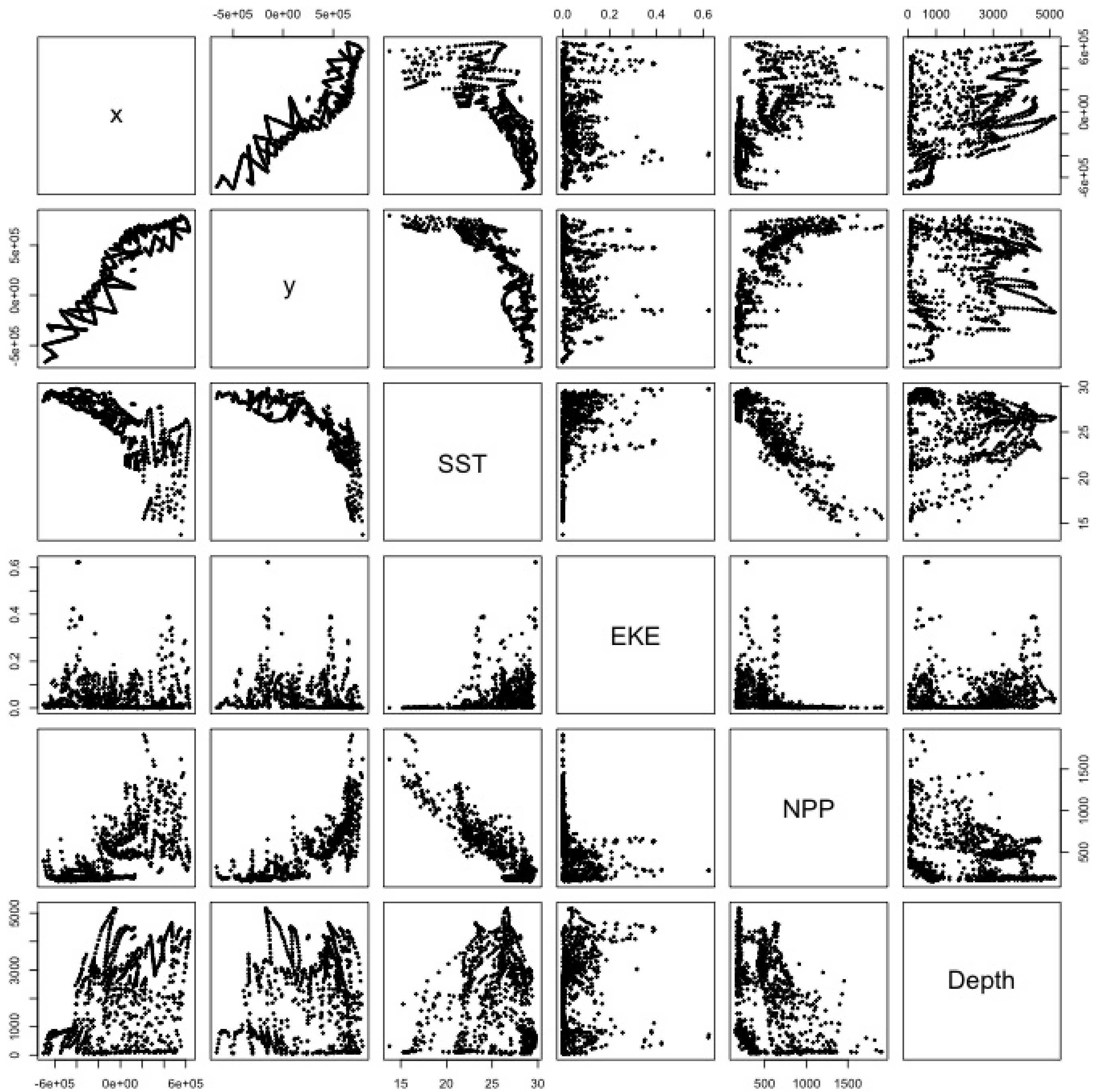


Tobler's first law of geography

"Everything is related to everything else, but near things are more related than distant things"

Tobler (1970)

Implications of Tobler's law



Covariates are not only correlated (linearly)…

…they are also “concurve”

“How much can one smooth be approximated by one or more other smooths?”

Concurvity (model/smooth)

```
concurvity(dsm_all)
```

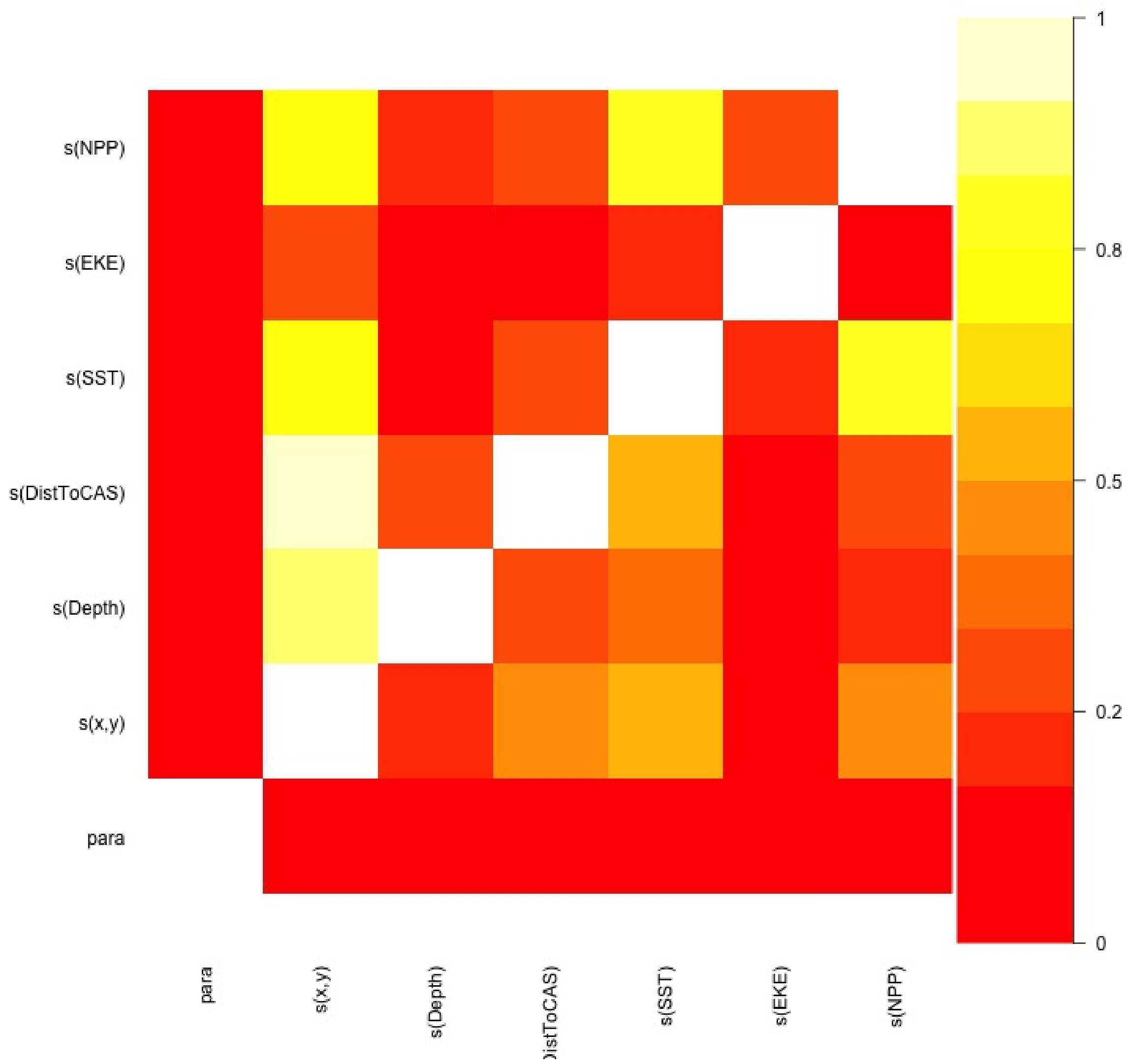
```
                 para    s(x,y)  s(Depth) s(DistToCAS)    s(SST)    s(EKE)
worst    2.539199e-23 0.9963493 0.9836597    0.9959057 0.9772853 0.7702479
observed 2.539199e-23 0.9213461 0.8275679    0.9883162 0.6951997 0.6615697
estimate 2.539199e-23 0.7580838 0.9272203    0.9642030 0.8978412 0.4906765
            s(NPP)
worst    0.9727752
observed 0.8258504
estimate 0.8694619
```

Concurvity between smooths

```
concurvity(dsm_all, full=FALSE)$estimate
```

```
                    para      s(x,y)    s(Depth) s(DistToCAS)
para        1.000000e+00 4.700364e-26 4.640330e-28 6.317431e-27
s(x,y)      8.687343e-24 1.000000e+00 9.067347e-01 9.568609e-01
s(Depth)    1.960563e-25 2.247389e-01 1.000000e+00 2.699392e-01
s(DistToCAS) 2.964353e-24 4.335154e-01 2.568123e-01 1.000000e+00
s(SST)      3.614289e-25 5.102860e-01 3.707617e-01 5.107111e-01
s(EKE)      1.283557e-24 1.220299e-01 1.527425e-01 1.205373e-01
s(NPP)      2.034284e-25 4.407590e-01 2.067464e-01 2.701934e-01
                  s(SST)      s(EKE)      s(NPP)
para        5.042066e-28 3.615073e-27 6.078290e-28
s(x,y)      7.205518e-01 3.201531e-01 6.821674e-01
s(Depth)    1.232244e-01 6.422005e-02 1.990567e-01
s(DistToCAS) 2.554027e-01 1.319306e-01 2.590227e-01
s(SST)      1.000000e+00 1.735256e-01 7.616800e-01
s(EKE)      2.410615e-01 1.000000e+00 2.787592e-01
s(NPP)      7.833972e-01 1.033109e-01 1.000000e+00
```

Visualising concurvity between terms



- Previous matrix output visualised
- High values (yellow) = BAD

Path dependence

Sensitivity

- General path dependency?
- What if there are highly concurve smooths?
- Is the model is sensitive to them?

What can we do?

- Fit variations excluding smooths
  - Concurve terms that are excluded early on
- Appendix of Winiarski et al (2014) has an example

Sensitivity example

- s(Depth) and s(x, y) are highly concurve (0.9067)
- Refit removing Depth first

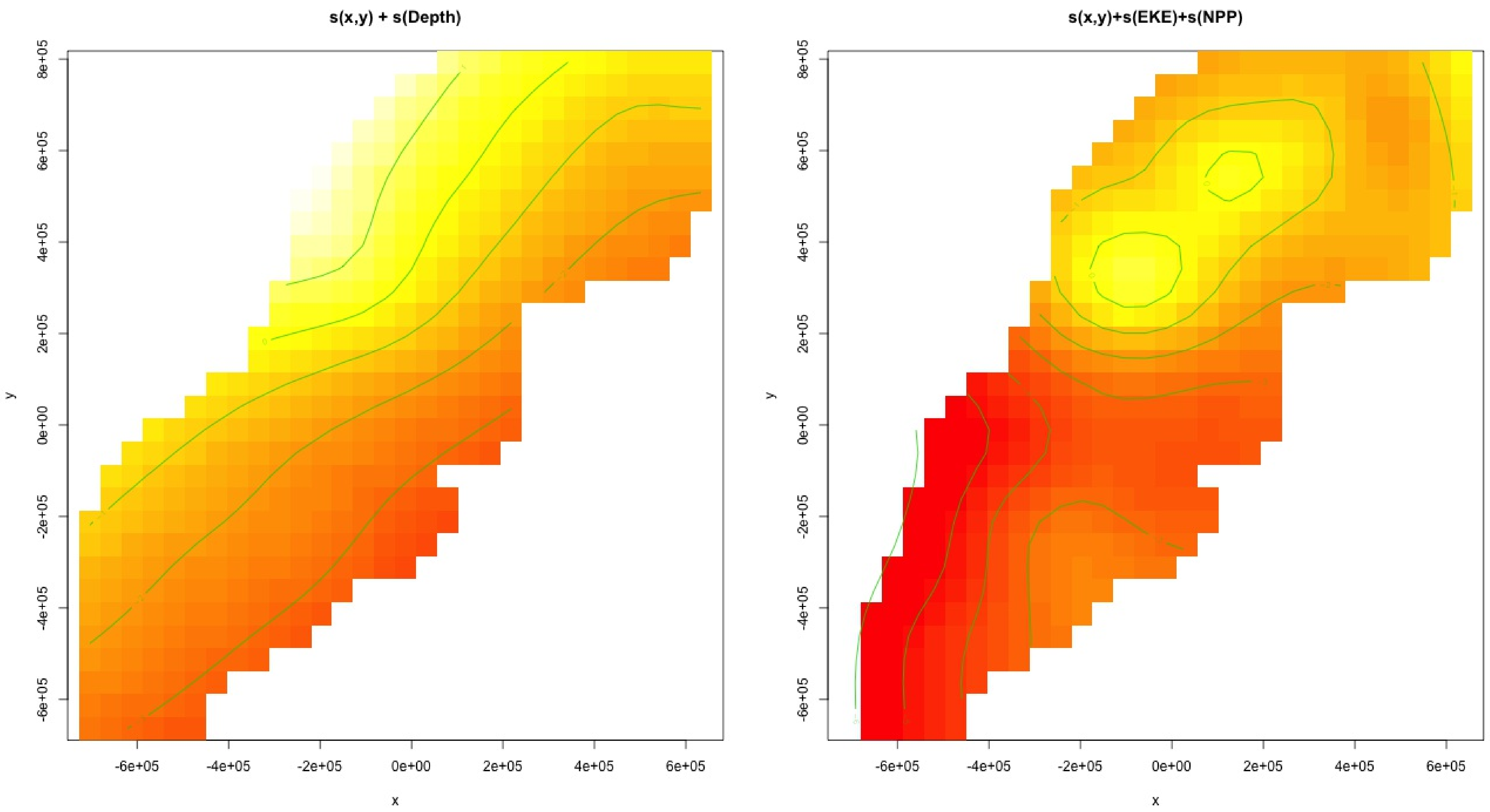
```
# with depth
```

```
              edf Ref.df        F     p-value
s(x,y)   6.443109     29 1.321664 4.75402e-08
s(Depth) 3.611031      9 4.261217 1.48593e-10
```

```
# without depth
```

```
              edf Ref.df        F      p-value
s(x,y) 13.7776636     29 2.589135 1.161592e-12
s(EKE)  0.8448449      9 0.566980 1.050411e-02
s(NPP)  0.7994187      9 0.362814 3.231808e-02
```

Comparison of spatial effects



Sensitivity example

- Refit removing x and y…

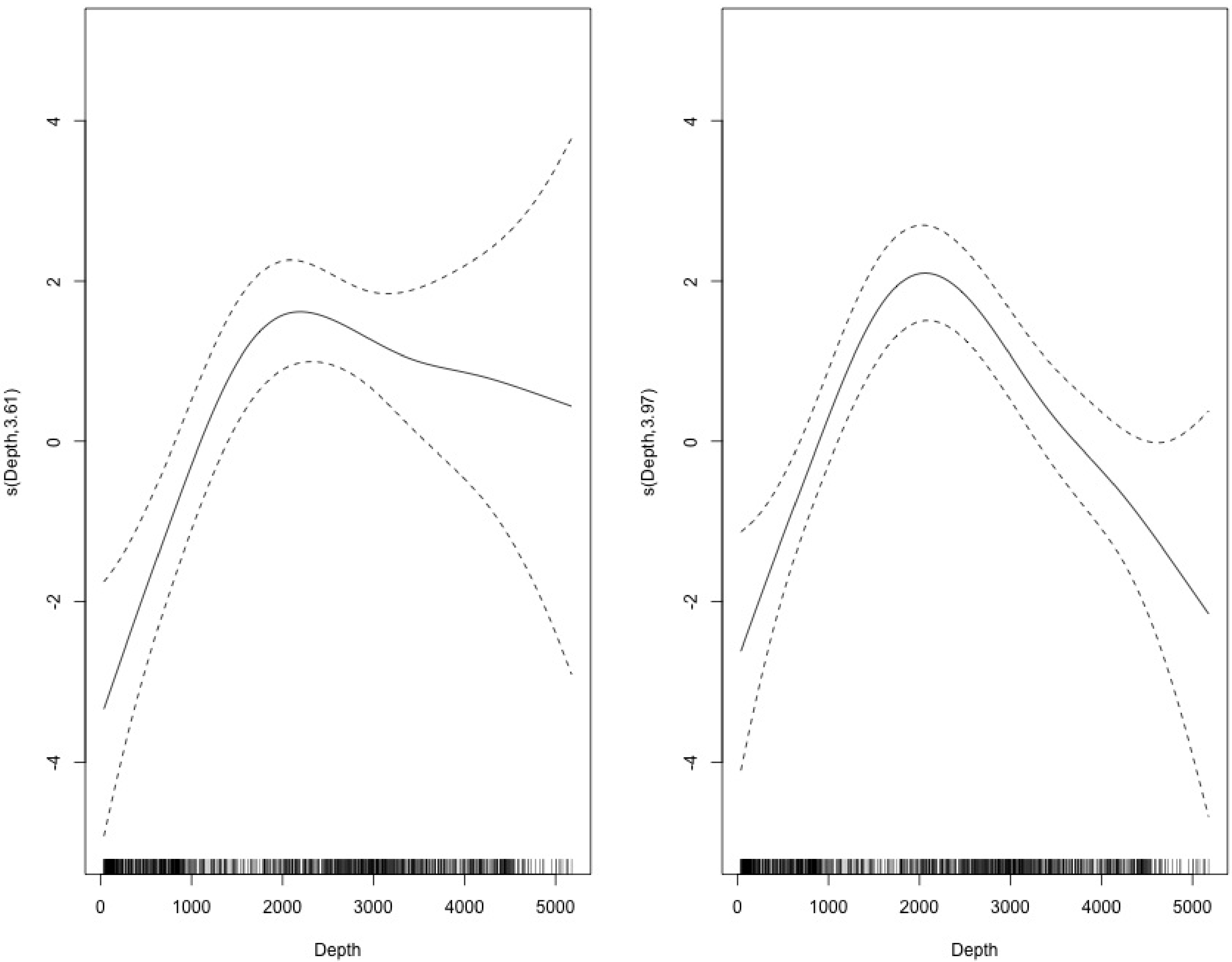
```
# without xy
```

```
             edf Ref.df        F      p-value
s(SST)   4.583260      9 3.244322 3.118815e-06
s(Depth) 3.973359      9 6.799043 4.125701e-14
```

```
# with xy
```

```
             edf Ref.df        F     p-value
s(x,y)   6.443109     29 1.321664 4.75402e-08
s(Depth) 3.611031      9 4.261217 1.48593e-10
```

Comparison of depth smooths



Comparing those three models...

Model	AIC	Deviance
s(x,y) + s(Depth)	1229.888	37.84
s(x,y)+s(EKE)+s(NPP)	1248.167	34.44
s(SST)+s(Depth)	1228.152	35.77

- "Full" model still explains most deviance
- No depth model requires spatial smooth to "mop up" extra variation
- We'll come back to this when we do prediction

Recap

Recap

- Adding smooths
- Removing smooths
  - p-values
  - shrinkage/extra penalties
- Comparing models
- Comparing response distributions
- Sensitivity