

# Lecture 4: Model checking



University of  
St Andrews

*"perhaps the most important part of applied statistical modelling"*

Simon Wood

# Model checking

- As with detection functions, checking is important
- Checking *doesn't* mean your model is **right**
- Want to know the model conforms to assumptions
- What assumptions should we check?

# Convergence

# Convergence

- Fitting the GAM involves an optimization
- By default this is REstricted Maximum Likelihood (REML) score
- Sometimes this can go wrong
- R will warn you!

# A model that converges

```
gam.check(dsm_tw_xy_depth)
```

```
##
## Method: REML    Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-3.456333e-05,1.051004e-05]
## (score 374.7249 & scale 4.172176).
## Hessian positive definite, eigenvalue range [1.179219,301.267].
## Model rank = 39 / 39
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'    edf k-index p-value
## s(x,y)    29.00 11.11    0.65 <2e-16 ***
## s(Depth)   9.00  3.84    0.81    0.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# A bad model

```
Error in while (mean(ldxx/(ldxx + ldss)) > 0.4) { :  
  missing value where TRUE/FALSE needed  
In addition: Warning message:  
In sqrt(w) : NaNs produced  
Error in while (mean(ldxx/(ldxx + ldss)) > 0.4) { :  
  missing value where TRUE/FALSE needed
```

This is **rare**

# The Folk Theorem of Statistical Computing

*"most statistical computational problems are due not to the algorithm being used but rather the model itself"*

Andrew Gelman



# Folk Theorem anecdota

- Often if there are fitting problems, you're asking too much from your data
- Model is too complicated
- Too little data
- Try something simpler, see what happens

Basis size

# Basis size (k)

- Set  $k$  per term
- e.g.  $s(x, k=10)$  or  $s(x, y, k=100)$
- Penalty removes "extra" wigglyness
  - *up to a point!*
- (But computation is slower with bigger  $k$ )

# Checking basis size

```
gam.check(dsm_x_tw)
```

```
##
## Method: REML    Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-3.196351e-06,4.485625e-07]
## (score 409.936 & scale 6.041307).
## Hessian positive definite, eigenvalue range [0.7645492,302.127].
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(x)  9.00  4.96    0.76    0.38
```

# Increasing basis size

```
dsm_x_tw_k <- dsm(count~s(x, k=20), ddf.obj=df,  
                  segment.data=segs, observation.data=obs,  
                  family=tw())  
gam.check(dsm_x_tw_k)
```

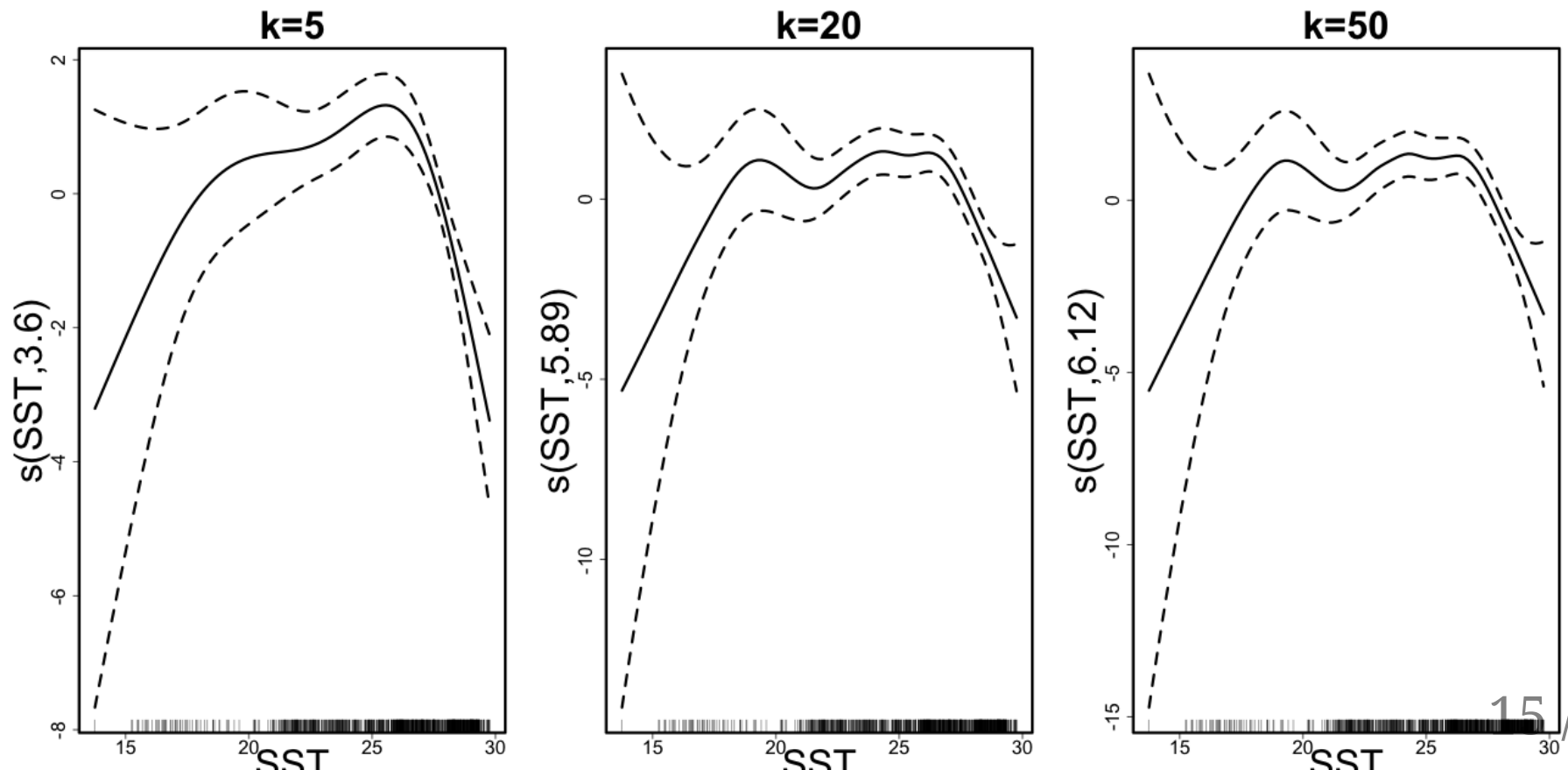
```
##  
## Method: REML    Optimizer: outer newton  
## full convergence after 7 iterations.  
## Gradient range [-2.30124e-08,3.930703e-09]  
## (score 409.9245 & scale 6.033913).  
## Hessian positive definite, eigenvalue range [0.7678456,302.0336].  
## Model rank = 20 / 20  
##  
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##  
##           k'    edf k-index p-value  
## s(x) 19.00  5.25    0.76    0.35
```

# Sometimes basis size isn't the issue...

- Generally, double  $k$  and see what happens
- Didn't increase the EDF much here
- Other things can cause low "p-value" and "k-index"
- Increasing  $k$  can cause problems (nullspace)

# k is a maximum

- Don't worry about things being too wiggly
- $k$  gives the maximum complexity
- Penalty deals with the rest



# Residuals



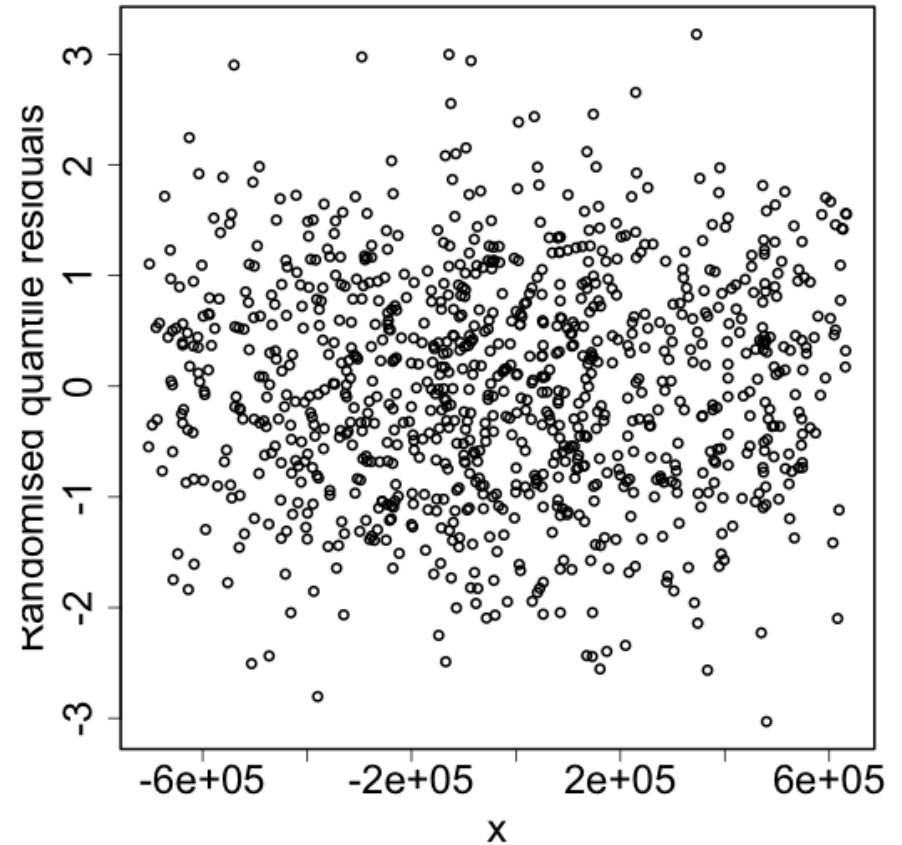
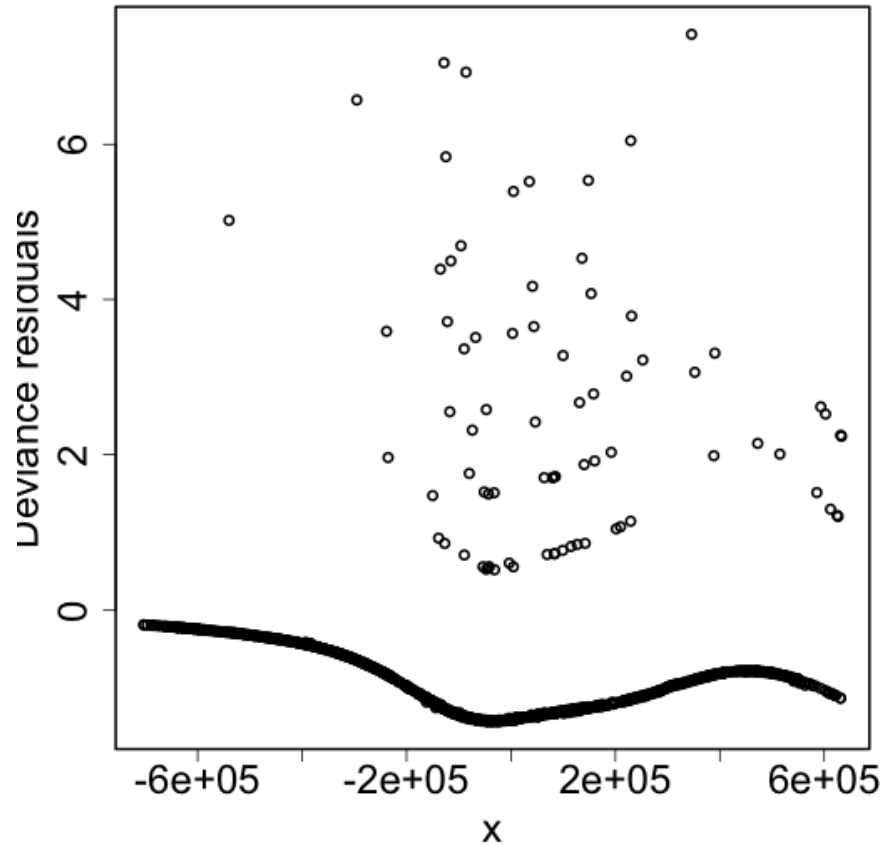
# What are residuals?

- Generally residuals = observed value - fitted value
- BUT hard to see patterns in these "raw" residuals
- Need to standardise  $\Rightarrow$  **deviance residuals**
- Expect these residuals  $\sim N(0, 1)$

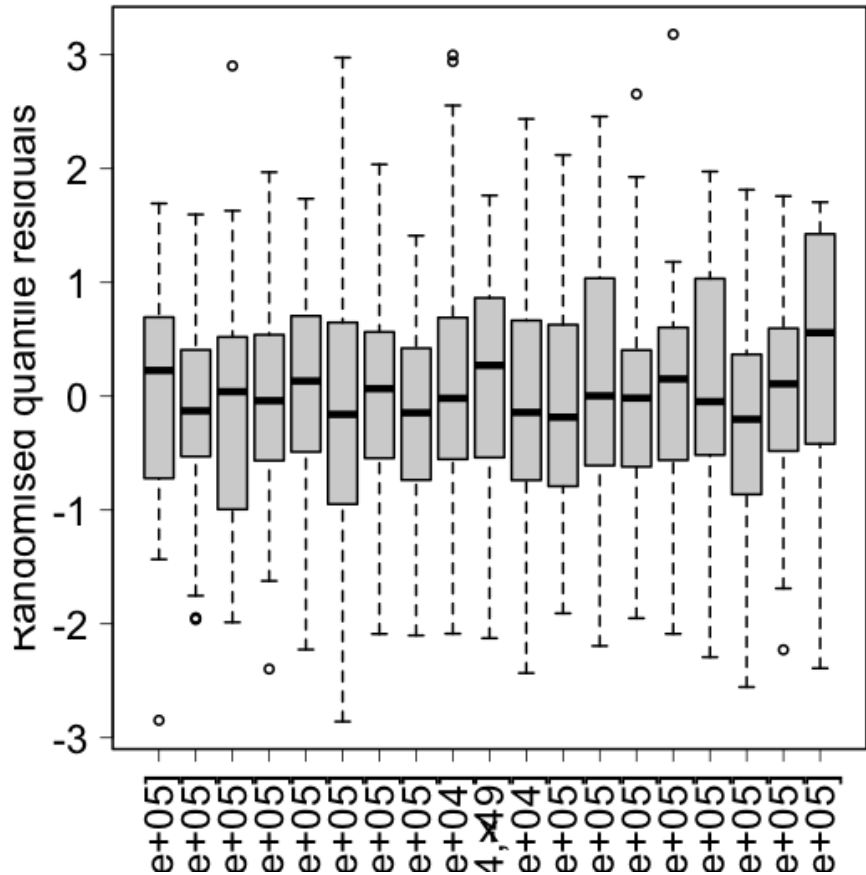
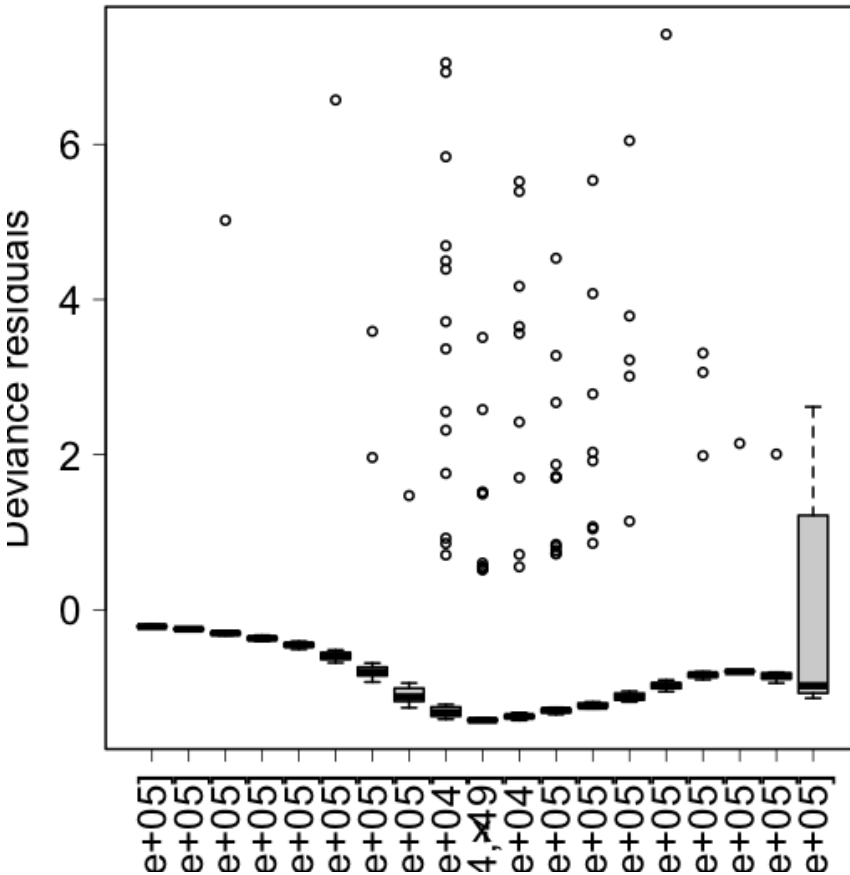
# Why are residuals important?

- Structure in the residuals means your model didn't capture something
- Maybe a missing covariate
- Model doesn't describe the data well

# Residuals vs. covariates



# Residuals vs. covariates (boxplots)



# Fitting to residuals

- Refit our model but with the residuals as response
- Response is normal (for deviance residuals)
- What pattern is left in the residuals?

# Example

- Example model with NPP and Depth

```
# get data
refit_dat <- dsm_depth_npp$data
# make residuals column
refit_dat$resid <- residuals(dsm_depth_npp)
# fit a model (same model)
resid_fit <- gam(resid~s(Depth, bs="ts", k=20) +
                  s(NPP, bs="ts", k=20),
                  family=gaussian(), data=refit_dat, method="REML")
```

# summary(resid\_fit)

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## resid ~ s(Depth, bs = "ts", k = 20) + s(NPP, bs = "ts", k = 20)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.49454    0.03274   -15.1   <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(Depth)  2.56621     19 1.230 4.9e-06 ***
## s(NPP)    0.03322     19 0.002  0.316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0241   Deviance explained = 2.67%
## -REML =    1362   Scale est. = 1.0174     n = 949
```

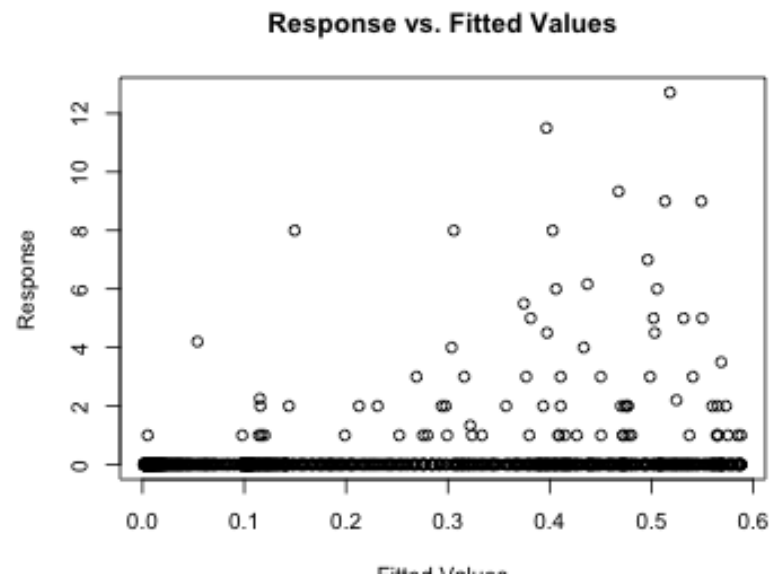
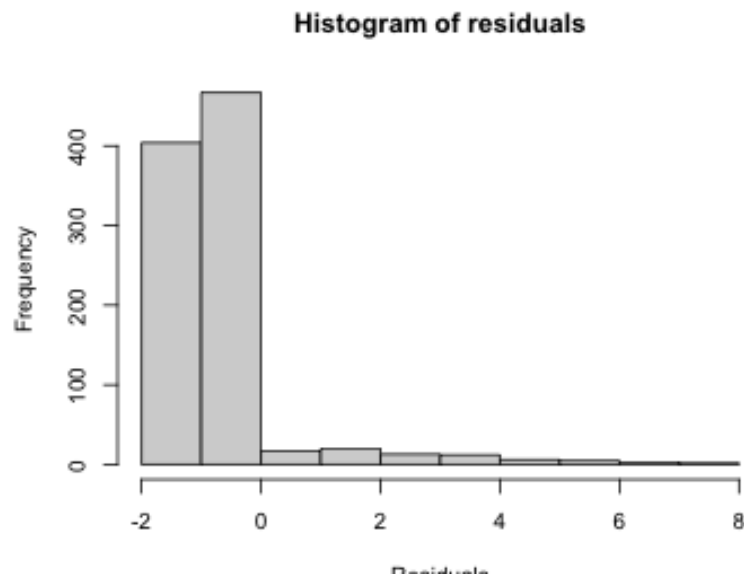
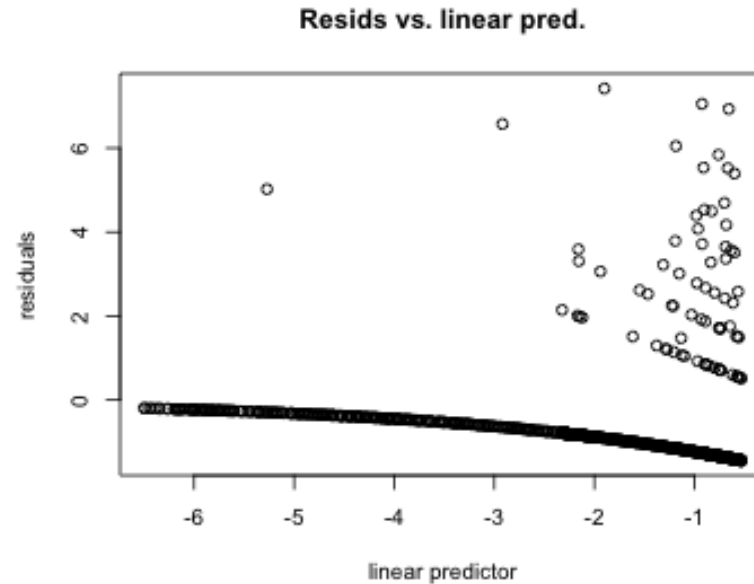
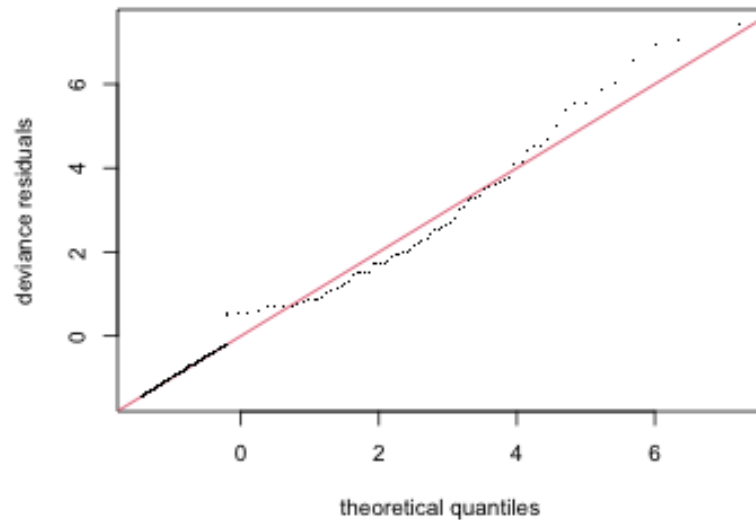
# What's going on there?

- Something unexplained going on?
- Maybe Depth + NPP is not enough?
  - Add other smooths ( $s(x, y)$ ?)
- Increase  $k$ ?



# Other residual checking

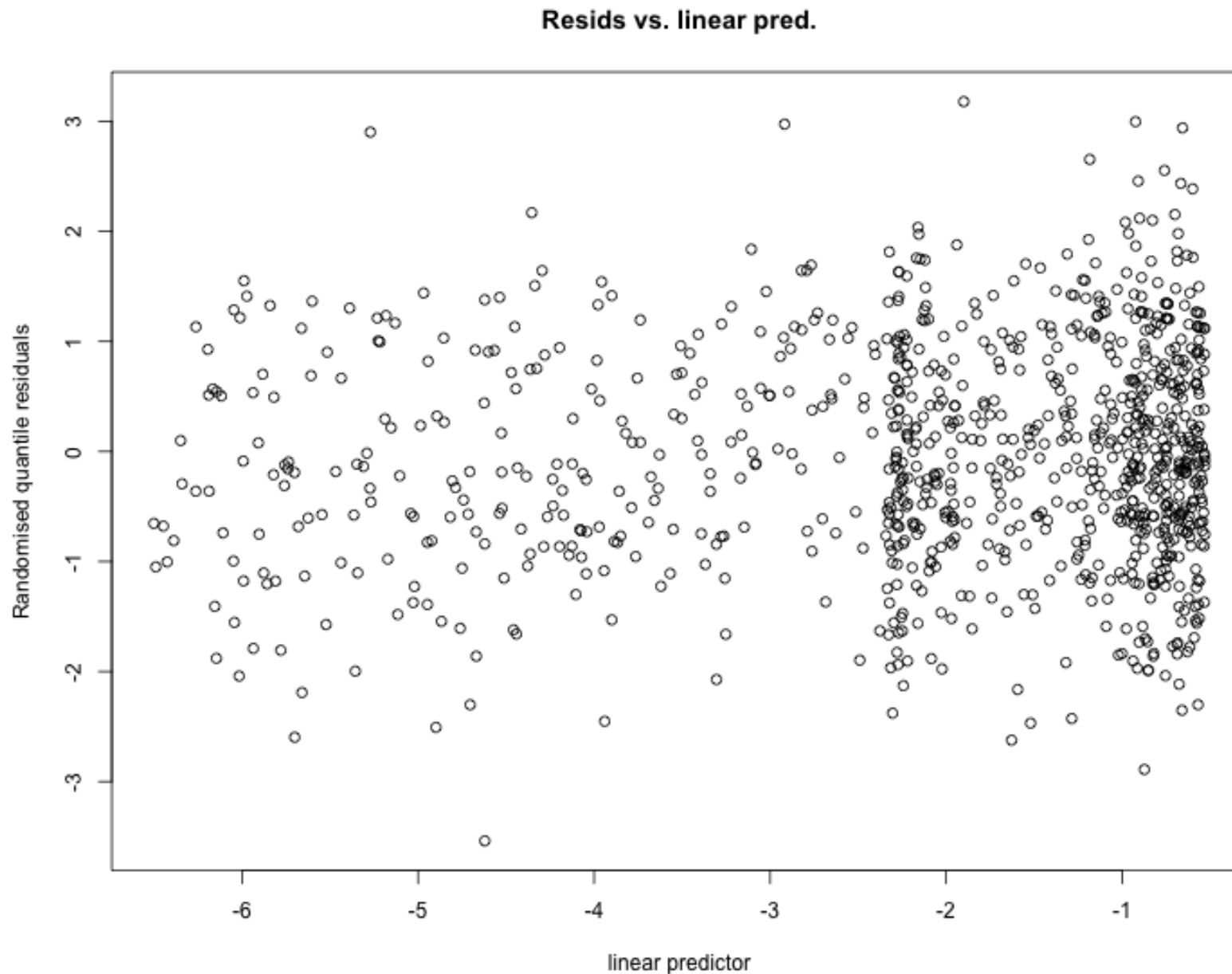
# gam.check



# Shortcomings

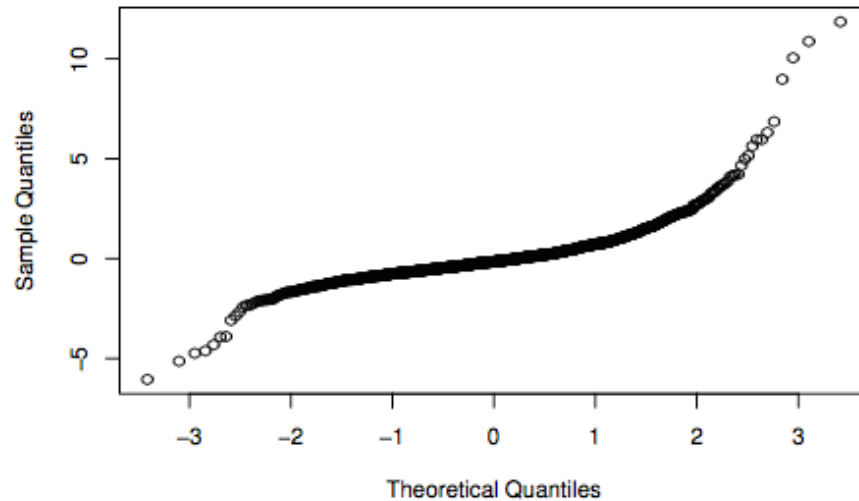
- `gam.check` can be helpful
- "Resids vs. linear pred" is victim of artifacts
- Need an alternative
- "Randomised quantile residuals"
  - `rqqam.check`
  - Exactly normal residuals

# Randomised quantile residuals

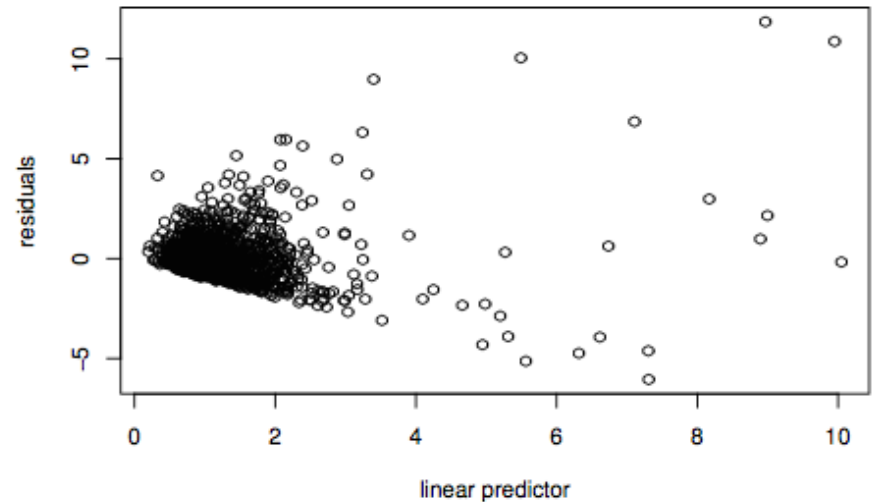


# Example of "bad" plots

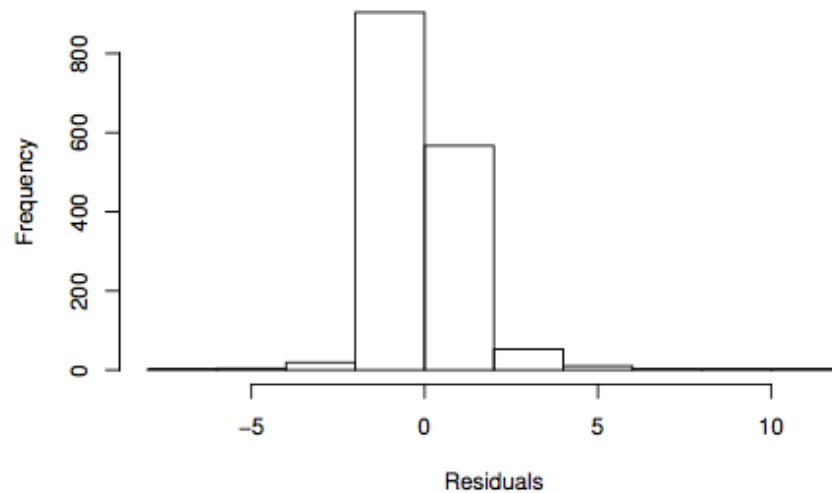
Normal Q-Q Plot



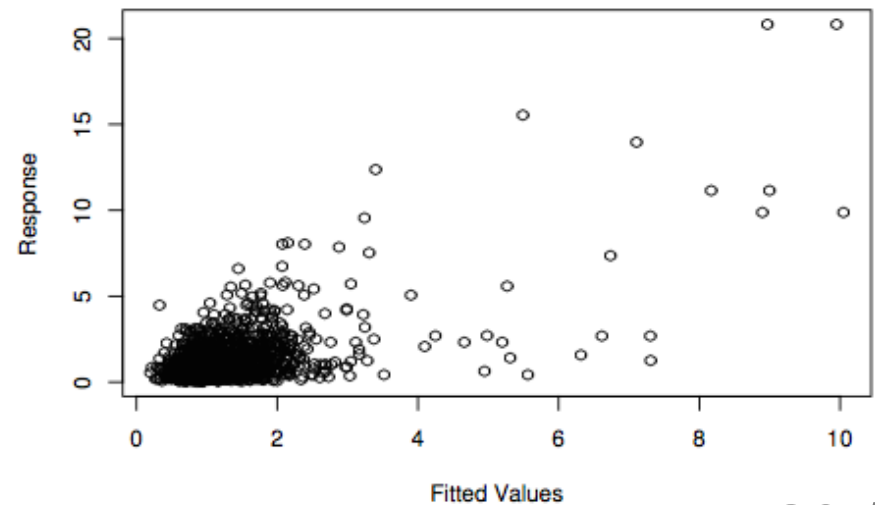
Resids vs. linear pred.



Histogram of residuals

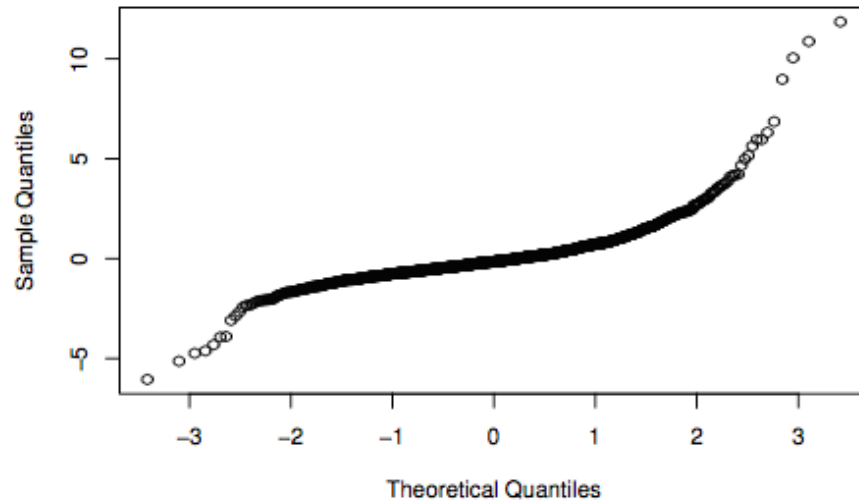


Response vs. Fitted Values

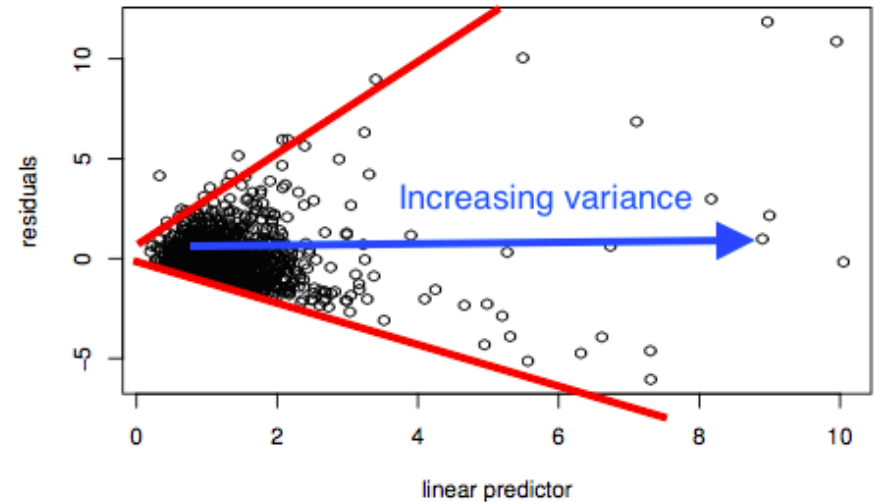


# Example of "bad" plots

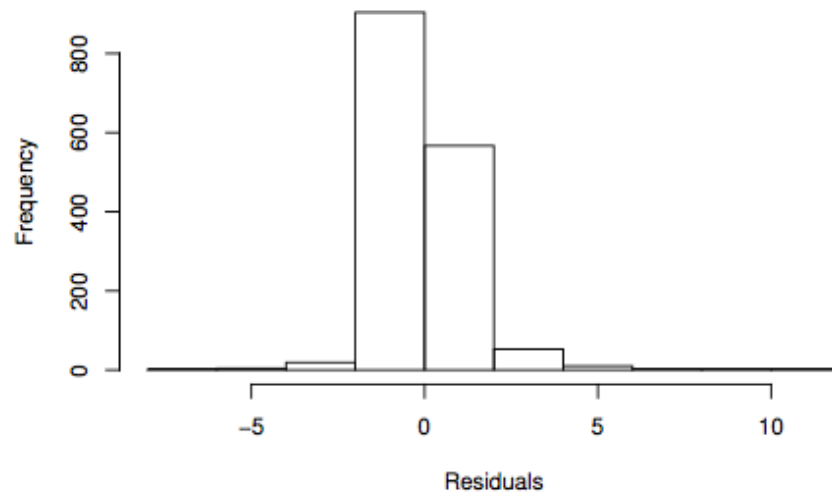
Normal Q-Q Plot



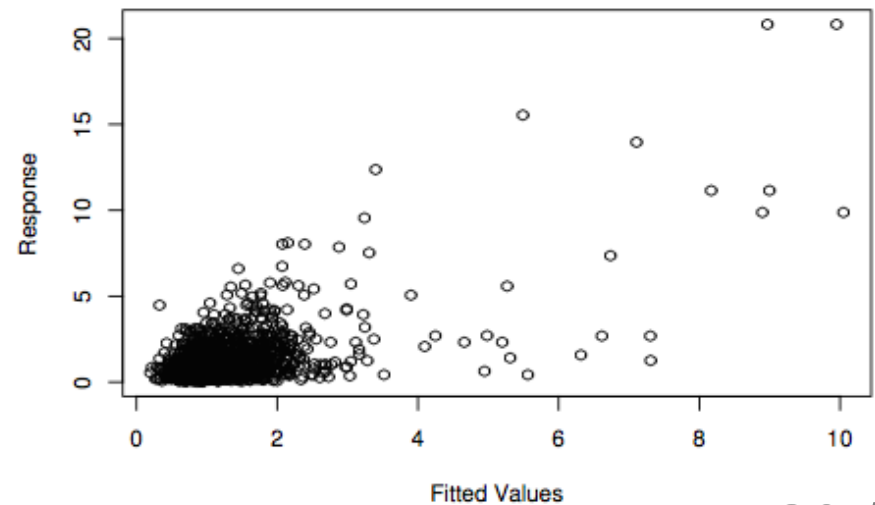
Resids vs. linear pred.



Histogram of residuals



Response vs. Fitted Values



# Residual checks

- Looking for patterns (not artifacts)
- This can be tricky
- Need to use a mixture of techniques
- Cycle through checks, make changes recheck

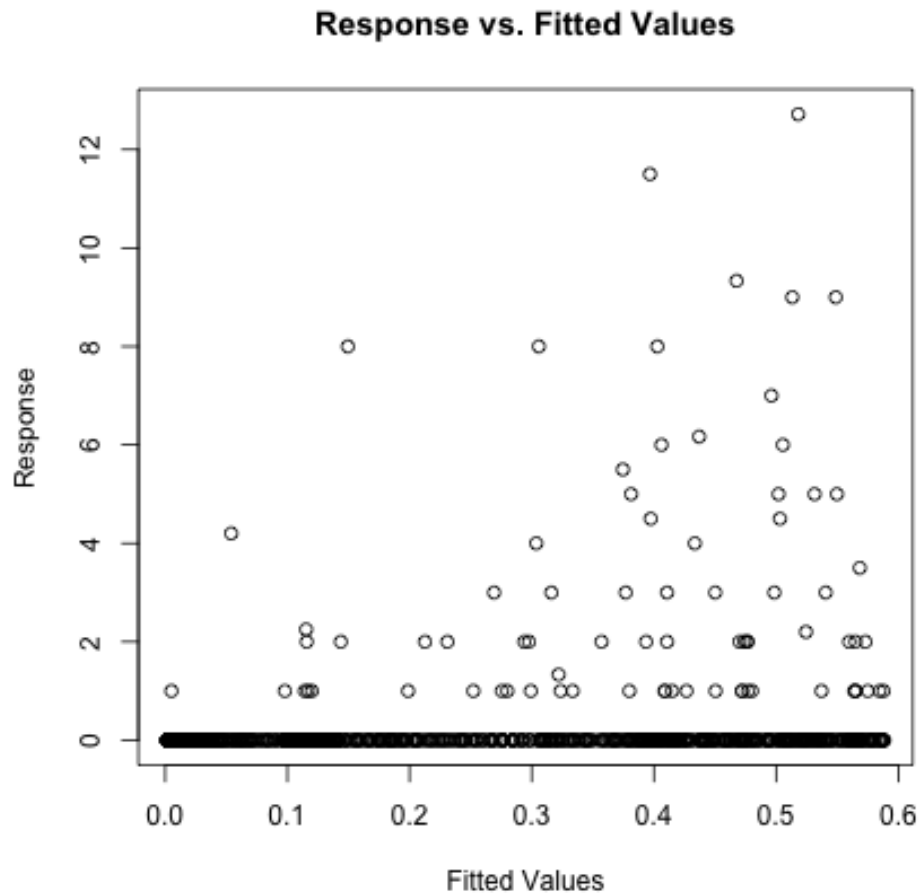
# Observed vs. expected

class: inverse, middle, center



# Response vs. fitted values

- `gam.check` "response vs. fitted values"
- BUT smooths are "wrong" everywhere in particular



# Summarize over covariate chunks

- On average the smooth is right
- Check aggregations of count
- Here detection function has Beaufort as factor

```
obs_exp(dsm_bad, "Beaufort_f")
```

```
##           [0,1]    (1,2]    (2,3]    (3,4]    (4,5]  
## Observed  1.00000  95.45000 103.5500  34.70000  4.000000  
## Expected 20.28781  54.57573 136.3581  53.98742  5.949304
```

```
obs_exp(dsm_good, "Beaufort_f")
```

```
##           [0,1]    (1,2]    (2,3]    (3,4]    (4,5]  
## Observed  1.00000  95.45000 103.5500  34.70000  4.000000  
## Expected  6.8887  45.18587 118.5747  53.81458  4.909644
```

# Observed vs. expected for environmental covariates

- Just need to specify the cutpoints

```
obs_exp(dsm_bad, "Depth", c(0, 1000, 2000, 3000, 4000, 6000))
```

##	(0,1e+03]	(1e+03,2e+03]	(2e+03,3e+03]	(3e+03,4e+03]	(4e+03,6e+03]
## Observed	4.000000	52.53333	139.16667	35.00000	8.000000
## Expected	85.65231	37.98341	63.40892	53.78726	30.32642

```
obs_exp(dsm_good, "Depth", c(0, 1000, 2000, 3000, 4000, 6000))
```

##	(0,1e+03]	(1e+03,2e+03]	(2e+03,3e+03]	(3e+03,4e+03]	(4e+03,6e+03]
## Observed	4.000000	52.53333	139.1667	35.00000	8.000000
## Expected	5.308628	48.14915	128.7962	38.76013	8.359456

# Summary

- Convergence
  - Rarely an issue
- Basis size
  - $k$  is a maximum
  - Double and see what happens
- Residuals
  - Deviance and randomised quantile
  - check for artifacts
- Observed vs. expected
  - Compare aggregate information