GAM case study: Cherry trees

STA303/1002 Winter 2022

This example comes from Wood (2016), full reference is the syllabus.

There is an optional video from last year talking through some of these functions here.

Set up

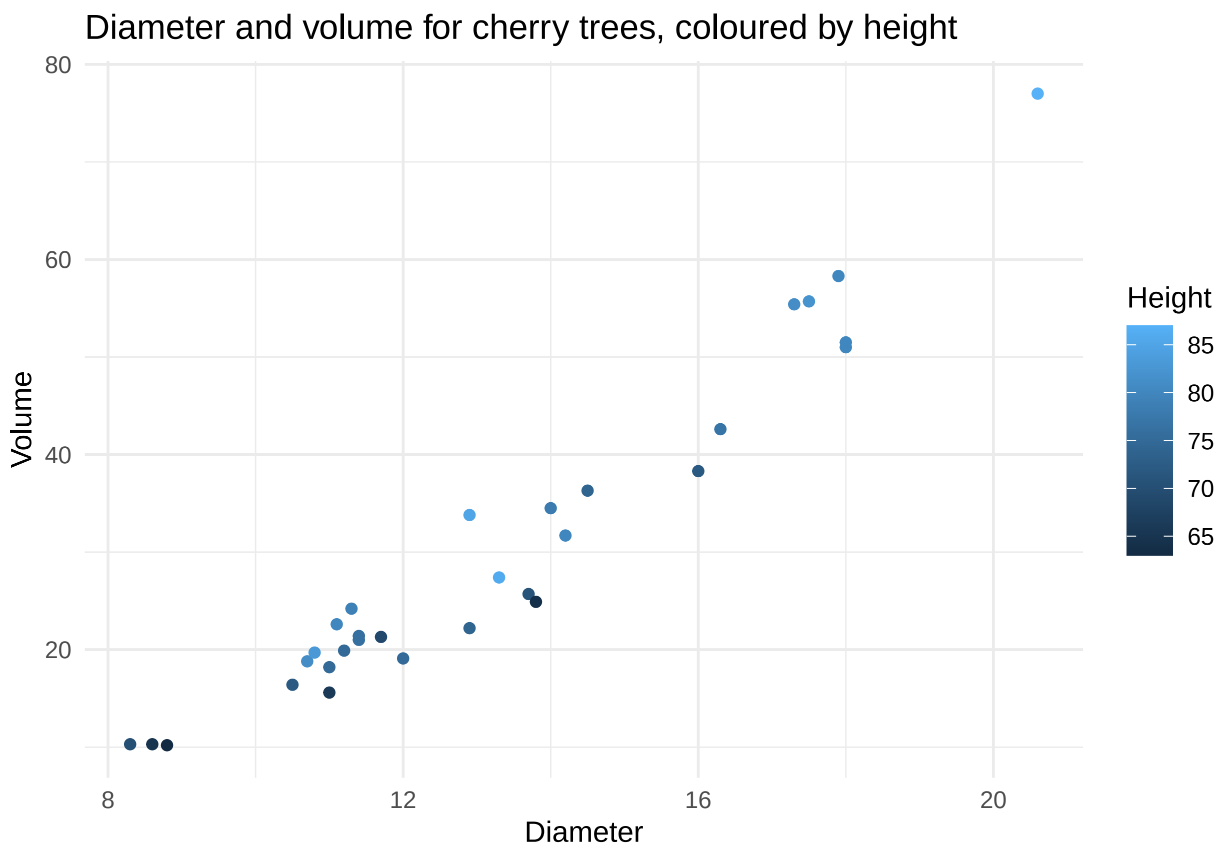
"This data set provides measurements of the diameter, height and volume of timber in 31 felled black cherry trees. Note that the diameter (in inches) is erroneously labelled Girth in the data. It is measured at 4 ft 6 in above the ground." -R documentation

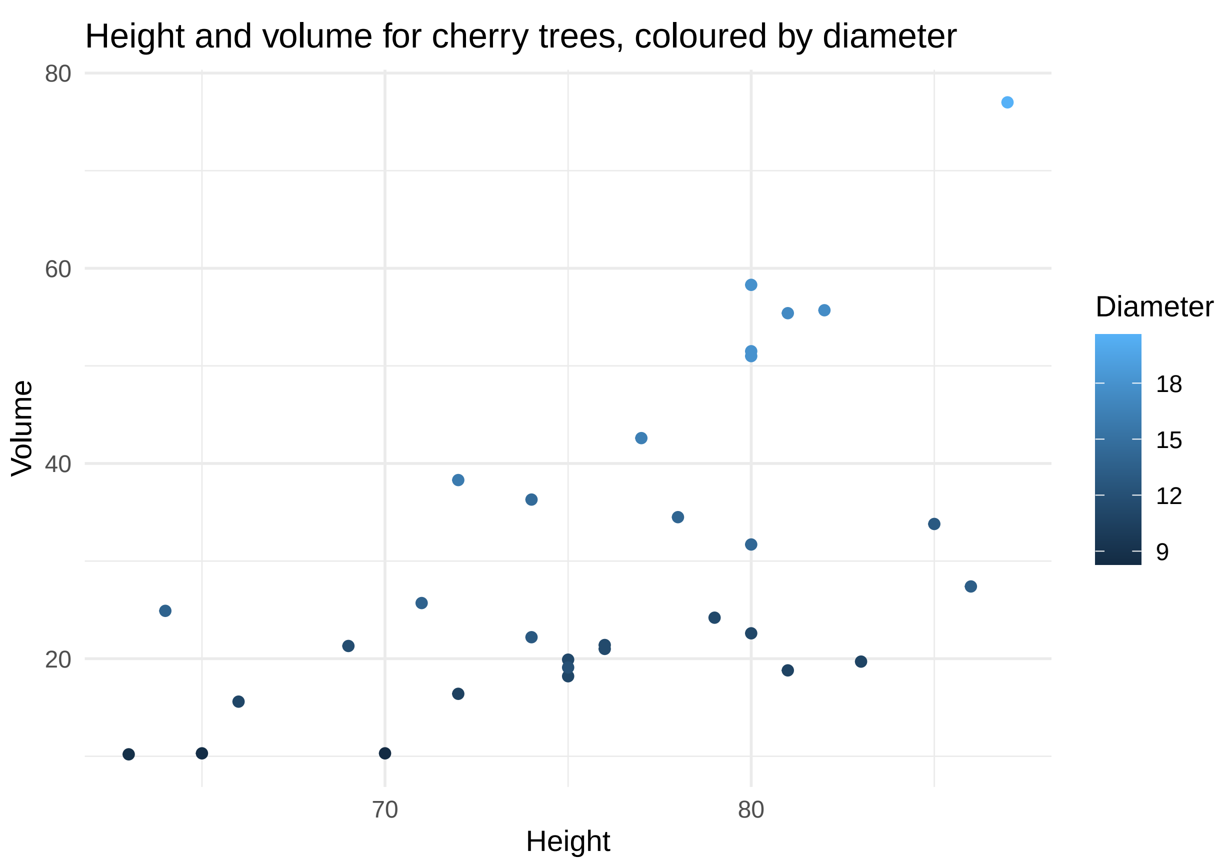
```
library(tidyverse)
library(MASS)
library(mgcv)
# install.packages("gratia")
library(gratia) # ggplot style
data('trees', package='datasets')
```

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

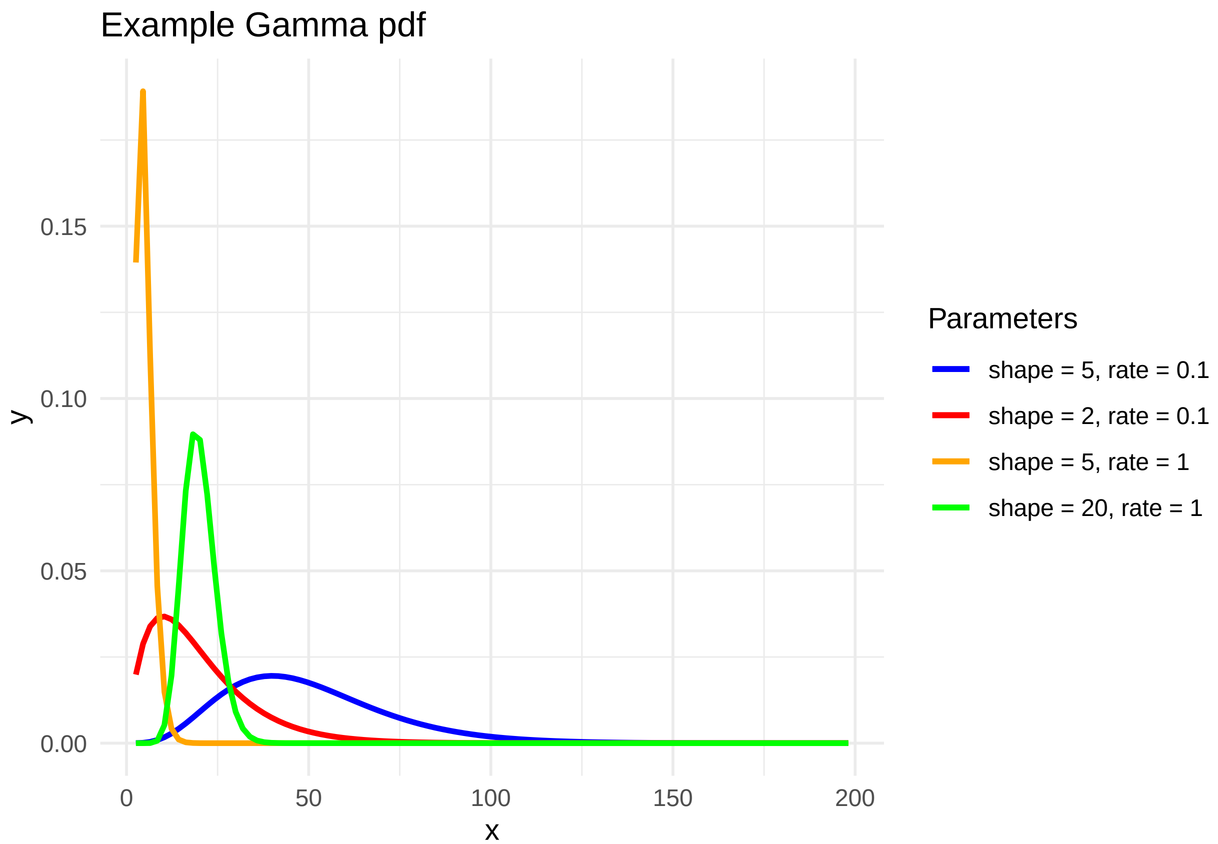
Exploratory visualizations





Aside: Gamma generalized linear models

We've met generalized linear models with Poisson and Logistic responses. Another fairly popular distribution for GLMs is Gamma. It's appropriate for when your response is > 0 and can handle right skew well. Log is *not* the **canonical link**, but is a popular choice.



Fitting the model

```
data(trees)

ct1<-gam(Volume~s(Height)+s(Girth), family=Gamma(link="log"),data=trees, select=TRUE, method="REML")
summary(ct1)
```

```
##
## Family: Gamma
## Link function: log
##
## Formula:
## Volume ~ s(Height) + s(Girth)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.27564    0.01484   220.7   <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(Height) 0.968      9  3.377 2.51e-06 ***
## s(Girth)  2.743      9 75.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.975   Deviance explained = 97.8%
## -REML = 77.736  Scale est. = 0.0068306  n = 31
```

This fits the model:

$$\log(\mathbb{E}[\text{Volume}_i]) = f_1(\text{Height}_i) + f_2(\text{Girth}_i)$$

$$\text{Volume}_i \sim \text{Gamma}(\alpha, \beta)$$

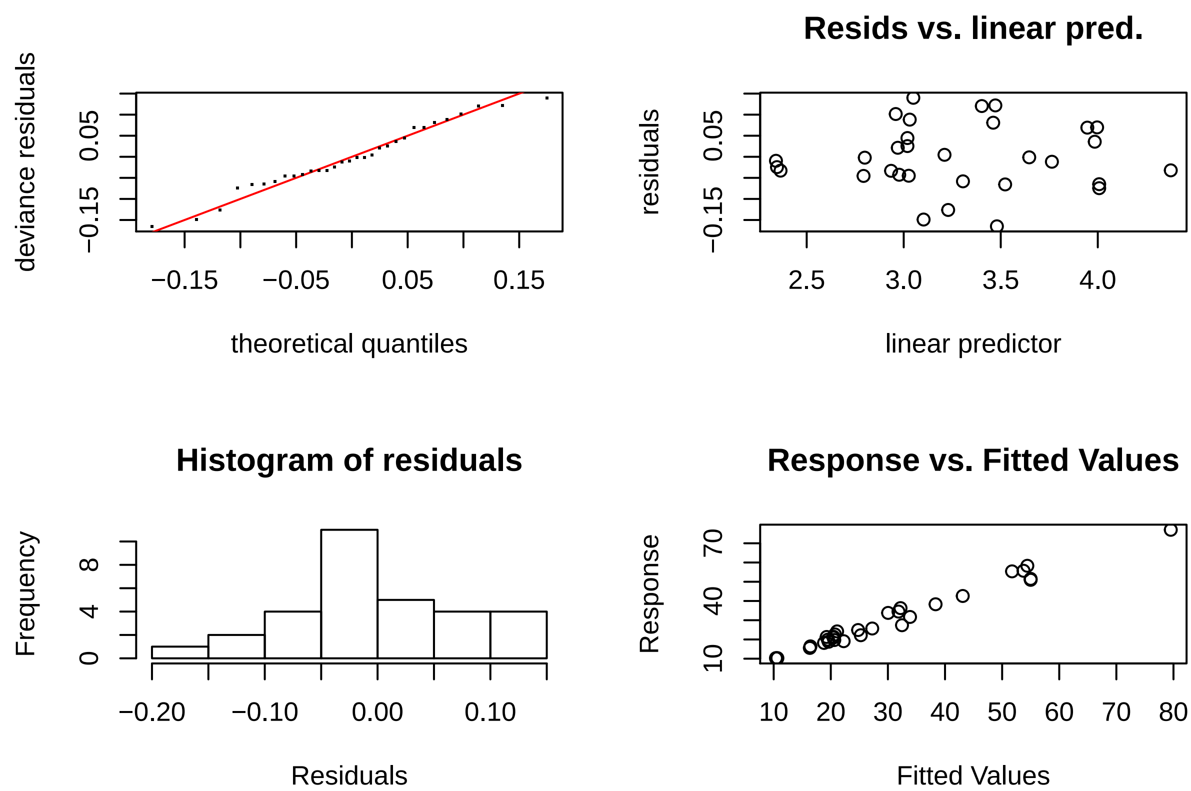
and `coef(ct1)` shows us the β_{jk}. Note: These β aren't interpretable in context.

```
coef(ct1)
```

```
##   (Intercept)   s(Height).1   s(Height).2   s(Height).3   s(Height).4
##  3.275642e+00 -3.368478e-07 -2.180055e-07 -6.015892e-08  1.613837e-07
##   s(Height).5   s(Height).6   s(Height).7   s(Height).8   s(Height).9
##  2.732841e-08  1.687224e-07 -5.721557e-08  8.622164e-07  9.879664e-02
##    s(Girth).1    s(Girth).2    s(Girth).3    s(Girth).4    s(Girth).5
##  2.027866e-02  5.396063e-02 -1.312387e-02 -3.697545e-02  1.633895e-02
##    s(Girth).6    s(Girth).7    s(Girth).8    s(Girth).9
##  3.386122e-02  4.391841e-03  2.241458e-01  4.322788e-01
```

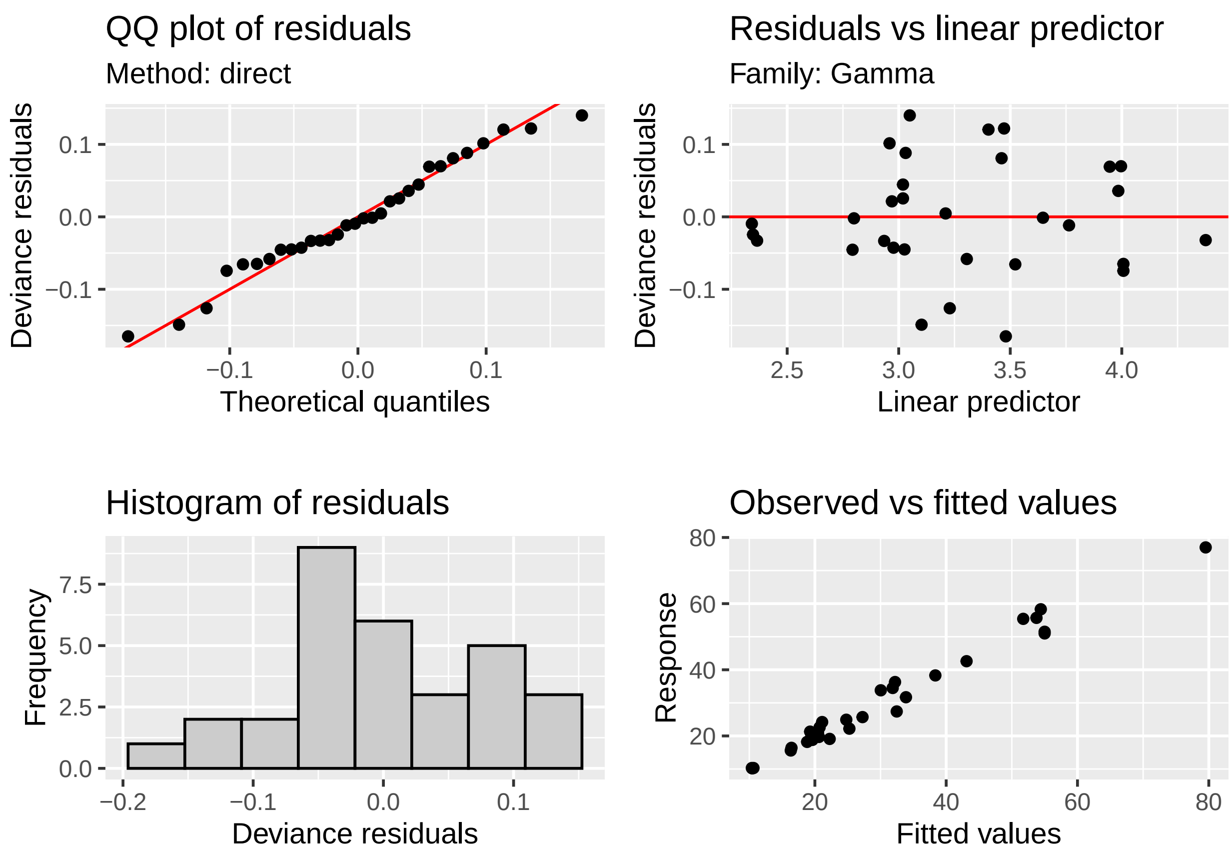
Diagnostics

```
# using gam.check
gam.check(ct1)
```

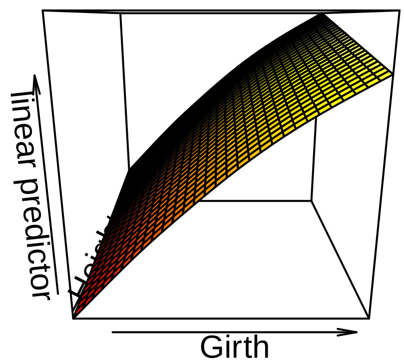


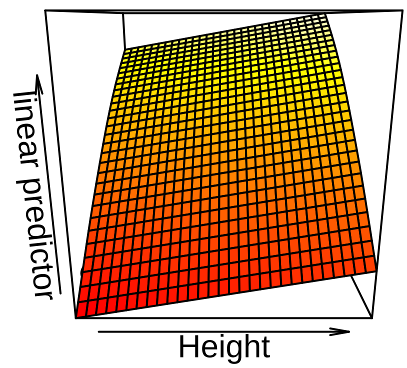

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 11 iterations.
## Gradient range [-4.225181e-06,9.677225e-06]
## (score 77.73553 & scale 0.006830568).
## Hessian positive definite, eigenvalue range [4.120126e-06,15.12282].
## Model rank =  19 / 19
##
## Basis dimension (k) 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(Height) 9.000 0.968    1.25    0.90
## s(Girth)  9.000 2.743    1.09    0.62
```

```
# same plots as above, but in a ggplot style
gratia::appraise(ct1)
```




```
par(mfrow=c(1,2))
vis.gam(ct1, view=c("Girth", "Height"))
vis.gam(ct1, view=c("Height", "Girth"))
```

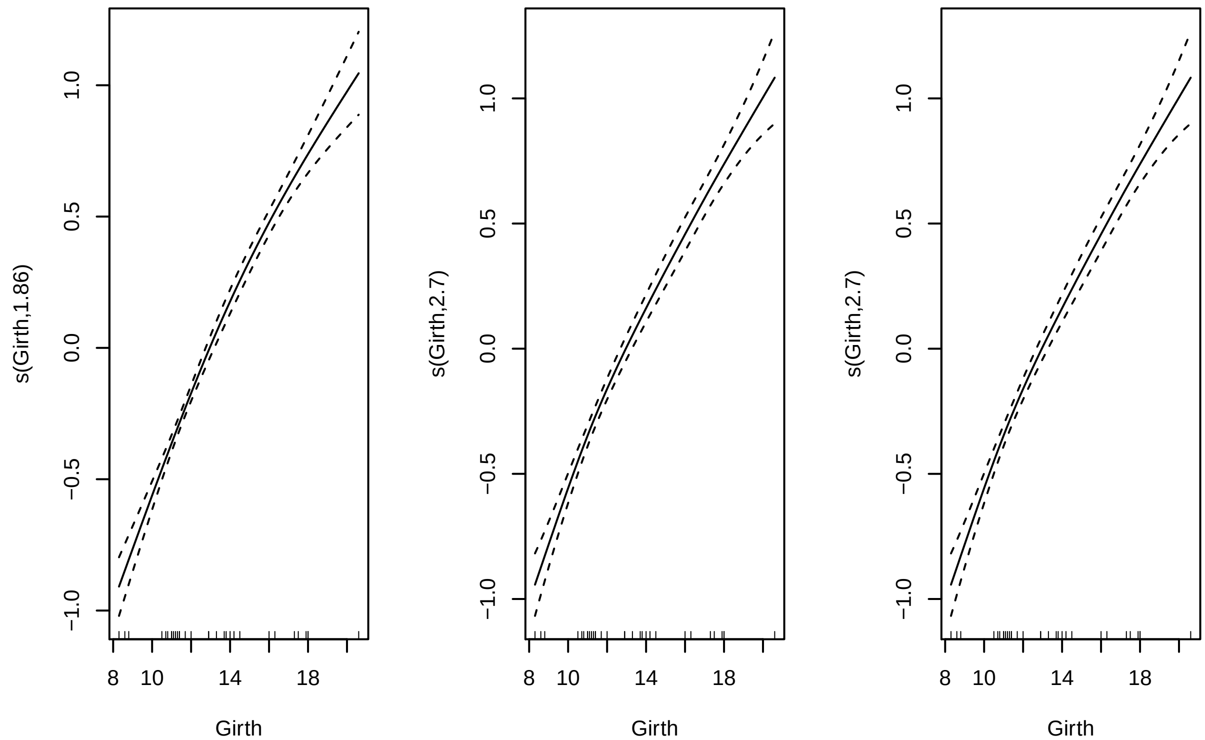




Basis dimension

```
res1 <-gam(Volume~s(Girth, k=3),
           family=Gamma(link="log"), data=trees)
res2 <-gam(Volume~s(Girth,k=15),
           family=Gamma(link="log"), data=trees)
res3 <-gam(Volume~s(Girth, k=25),
           family=Gamma(link="log"), data=trees)

par(mfrow=c(1, 3))
plot(res1)
plot(res2)
plot(res3)
```



$\hat{f}$ is smooth, don't need many basis functions

Testing

If you want to perform LR tests, you should probably use ML as the smoothing selection method and not use `select=TRUE` as the approximation used can be very bad for smooths with penalties on their null spaces. This treats our smooths as random effects.

```
ct1_ml <- gam(Volume~s(Height) + s(Girth), family=Gamma(link="log"), data=trees, method="ML")
ct2_ml <- gam(Volume ~ Height + s(Girth),  family=Gamma(link="log"),  data = trees, method = "ML")
ct3_ml <- gam(Volume ~ s(Girth), family=Gamma(link="log"), data = trees, method = "ML")

lmtest::lrtest(ct1_ml, ct2_ml)
```

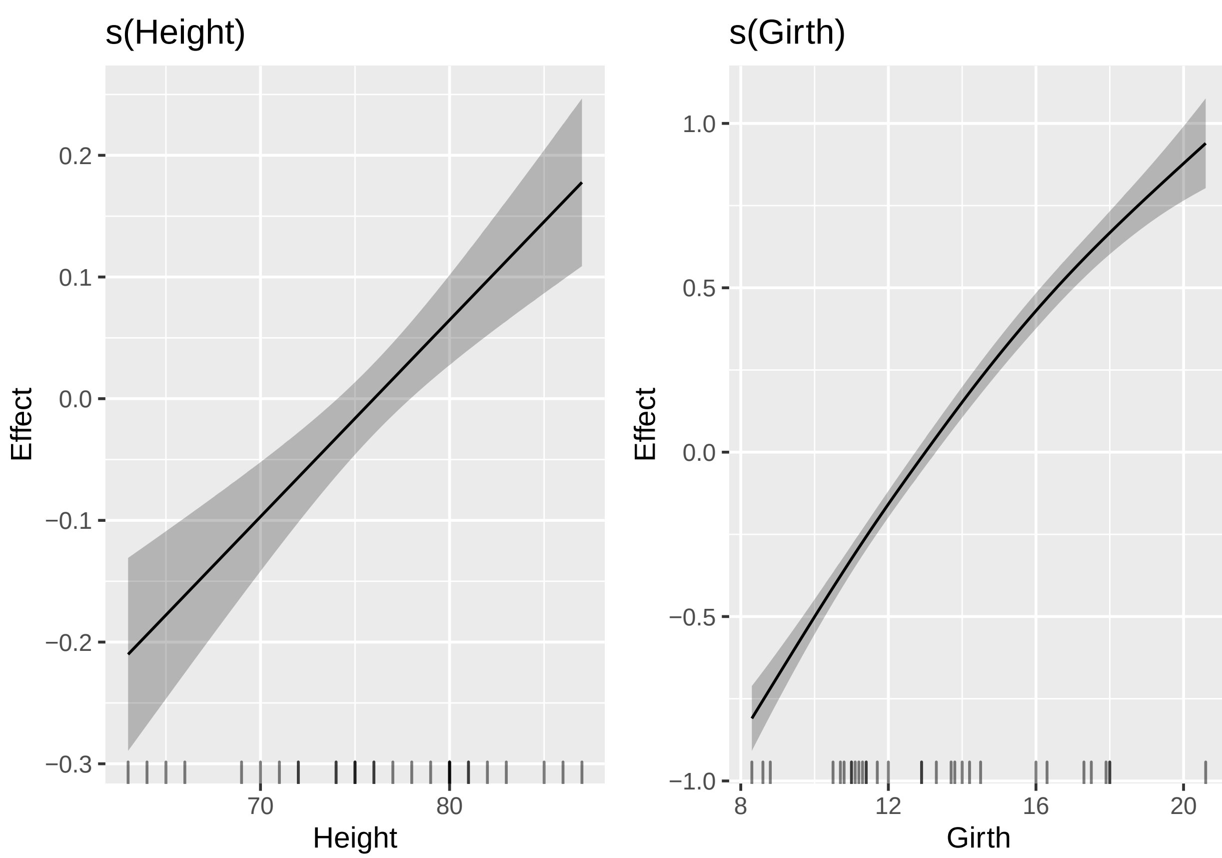
```
## Likelihood ratio test
##
## Model 1: Volume ~ s(Height) + s(Girth)
## Model 2: Volume ~ Height + s(Girth)
##      #Df  LogLik          Df Chisq Pr(>Chisq)
## 1 6.1496 -65.909
## 2 6.1495 -65.909 -3.6097e-05     0  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmtest::lrtest(ct2_ml, ct3_ml)
```

```
## Likelihood ratio test
##
## Model 1: Volume ~ Height + s(Girth)
## Model 2: Volume ~ s(Girth)
##      #Df  LogLik      Df  Chisq Pr(>Chisq)
## 1 6.1495 -65.909
## 2 5.0287 -77.552 -1.1209 23.286  1.396e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can use the `draw()` function to make comments about these variables.

```
draw(ct1_ml)
```



Predictions

`predict.gam` allows us to make predictions from our fitted models in the way we're used to from `predict`.

```
trees$pred <- predict(ct1, type="response")

trees %>%
```

```
ggplot(aes(Volume, pred)) +
geom_point() +
geom_abline(intercept = 0, slope = 1) +
theme_minimal() +
labs(x = "Observed volume", y = "Predicted volume")
```