An introduction to Bayesian modelling with brms and Stan

Francisco Rodríguez-Sánchez

https://frodriguezsanchez.net

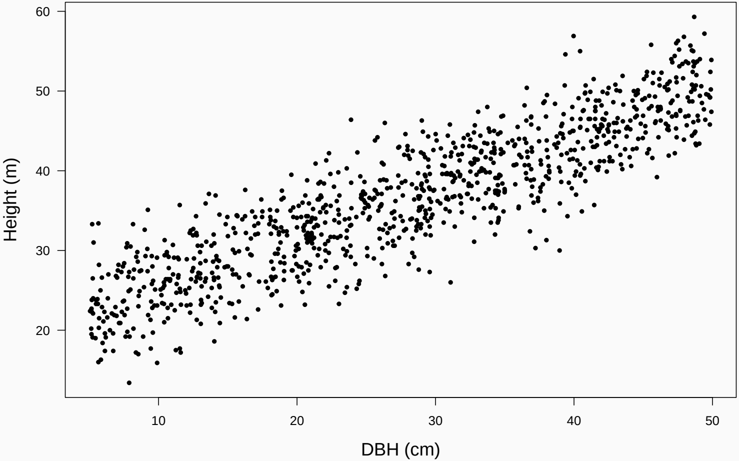
Our dataset: tree heights and DBH

- One species
- 10 plots
- 1000 trees
- Number of trees per plot ranging from 4 to 392

```
trees <- read.csv('data/trees.csv')
```

```
      site            dbh            height     
 Min.   : 1.0   Min.   : 5.06   Min.   :13.40  
 1st Qu.: 1.0   1st Qu.:17.69   1st Qu.:29.68  
 Median : 2.0   Median :28.62   Median :36.55  
 Mean   : 2.7   Mean   :27.88   Mean   :36.51  
 3rd Qu.: 4.0   3rd Qu.:38.97   3rd Qu.:43.33  
 Max.   :10.0   Max.   :49.92   Max.   :59.30  
```

What’s the relationship between DBH and height?



First step: linear regression (lm)

```
simple.lm <- lm(height ~ dbh, data = trees)
```

```

Call:
lm(formula = height ~ dbh, data = trees)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.3270  -2.8978   0.1057   2.7924  12.9511 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 19.33920    0.31064   62.26   <2e-16 ***
dbh          0.61570    0.01013   60.79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.093 on 998 degrees of freedom
Multiple R-squared:  0.7874,    Adjusted R-squared:  0.7871 
F-statistic:  3695 on 1 and 998 DF,  p-value: < 2.2e-16
```

Center continuous variables

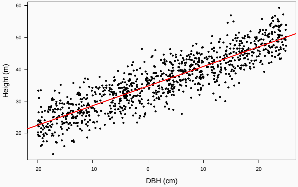
```
summary(trees$dbh)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.06   17.69   28.62   27.88   38.97   49.92
```

```
trees$dbh.c <- trees$dbh - 25
```

So, all parameters will be referred to a 25 cm DBH tree.

Linear regression with centred DBH



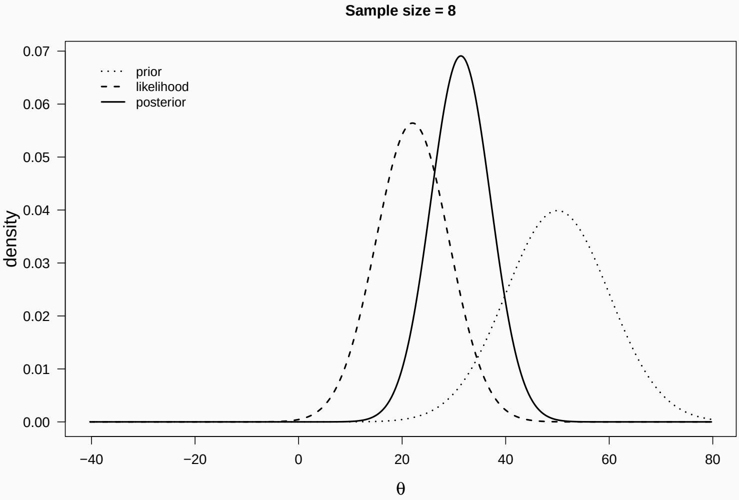
```
lm(formula = height ~ dbh.c, data = tree
            coef.est coef.se
(Intercept) 34.73    0.13
dbh.c        0.62    0.01
---
n = 1000, k = 2
residual sd = 4.09, R-Squared = 0.79
```

Let’s make it Bayesian

Bayesian inference: prior, posterior, and likelihood

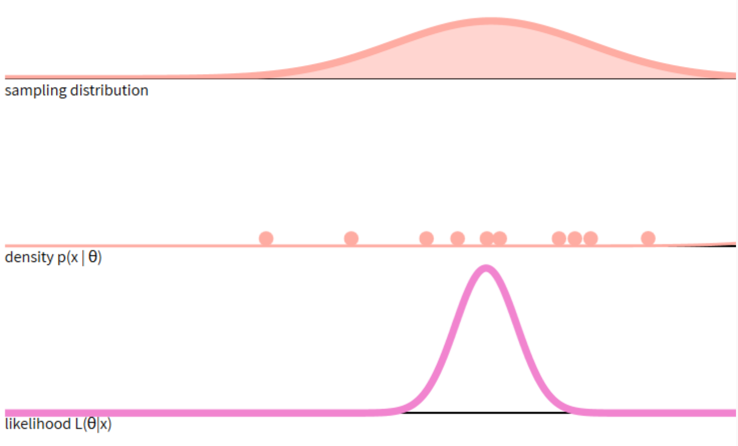
$$P(Unknown|Data) \propto P(Data|Unknown) \times P(Unknown)$$

$$Posterior \propto Likelihood \times Prior$$



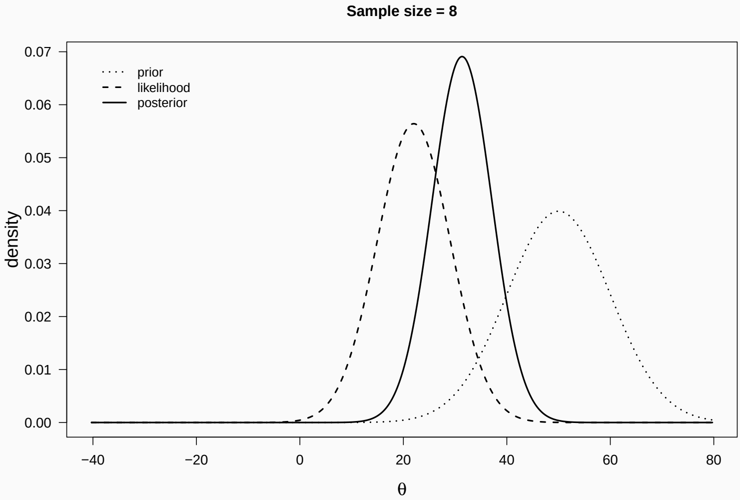
What is the likelihood?

$$L(\theta|x) = P(x|\theta)$$

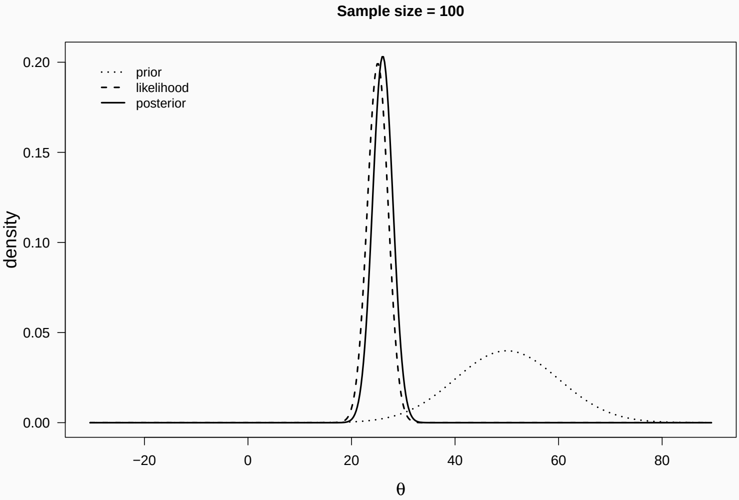


https://seeing-theory.brown.edu/bayesian-inference/index.html

Bayesian inference: prior and likelihood produce posterior



With increasing sample size, likelihood dominates prior



Bayesian statistics in practice

- Integrate information (**prior**)

Bayesian statistics in practice

- Integrate information (**prior**)
- Prior **regularises** unlikely estimates from data

Bayesian statistics in practice

- Integrate information (**prior**)
- Prior **regularises** unlikely estimates from data
- Particularly important with **limited** sample sizes

Bayesian statistics in practice

- Integrate information (**prior**)
- Prior **regularises** unlikely estimates from data
- Particularly important with **limited** sample sizes
- Large dataset -> prior effect **diminishes**

Bayesian statistics in practice

- Integrate information (**prior**)
- Prior **regularises** unlikely estimates from data
- Particularly important with **limited** sample sizes
- Large dataset -> prior effect **diminishes**
- **Uncertainty** / **Propagate errors**

Remember our model structure

$$y_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = \alpha + \beta x_i$$

In this case:

$$Height_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = \alpha + \beta DBH_i$$

α: expected height when DBH = 25 cm

β: how much height increases with every unit increase of DBH

Defining model formula

```r
library('brms')

height.formu <- brmsformula(height ~ dbh.c)
```

We must define **prior distributions** for every parameter

brms default priors

```r
get_prior(height.formu, data = trees)
```

```
                  prior     class  coef group resp dpar nlpar lb ub tag
                 (flat)         b                                      
                 (flat)         b dbh.c                                
 student_t(3, 36.5, 10.2) Intercept                                    
   student_t(3, 0, 10.2)     sigma                              0      
       source
      default
 (vectorized)
      default
      default
```

Choosing priors

Avoid ‘non-informative’ priors

Use *weakly informative* (e.g. relatively wide Normal or t-student distributions)

or *strongly informative* priors based on previous knowledge and common sense.

Some tips for setting priors:

- `https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations`

Run **prior predictive checks** (just priors, no data)

Choosing priors

Avoid ‘non-informative’ priors

Use *weakly informative* (e.g. relatively wide Normal or t-student distributions)

or *strongly informative* priors based on previous knowledge and common sense.

Some tips for setting priors:

- `https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations`
- Priors chapter in The BUGS book

Run **prior predictive checks** (just priors, no data)

Choosing priors

Avoid ‘non-informative’ priors

Use *weakly informative* (e.g. relatively wide Normal or t-student distributions)

or *strongly informative* priors based on previous knowledge and common sense.

Some tips for setting priors:

- `https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations`
- Priors chapter in The BUGS book
- `https://doi.org/10.1111/oik.05985`

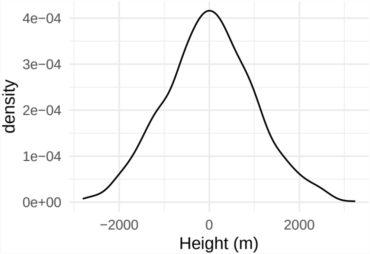
Run **prior predictive checks** (just priors, no data)

Probability distribution explorer

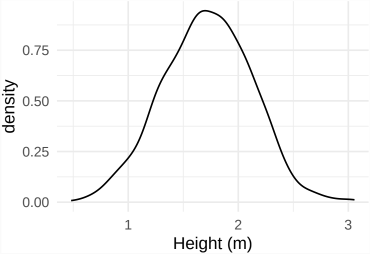
https://distribution-explorer.github.io/

Example: estimating people height across countries

Unreasonable prior



Reasonable prior



Defining priors for our trees example

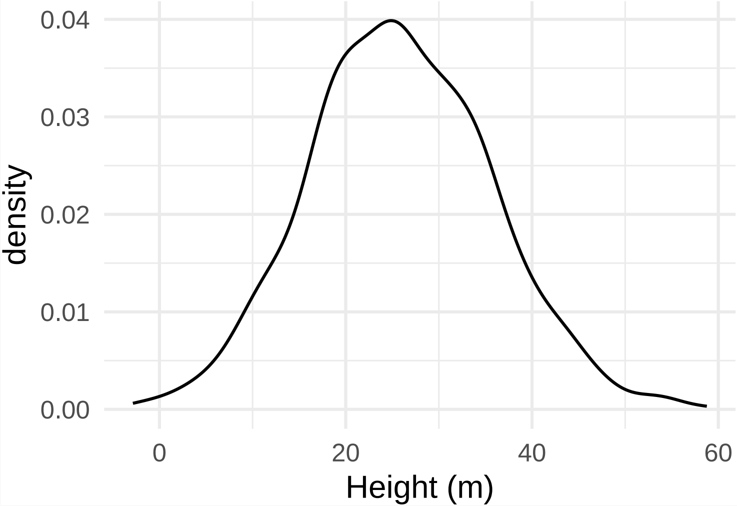
Intercept: Expected height of 25-cm DBH tree

b (slope): How much tree height varies (in m) when DBH increases 1 cm

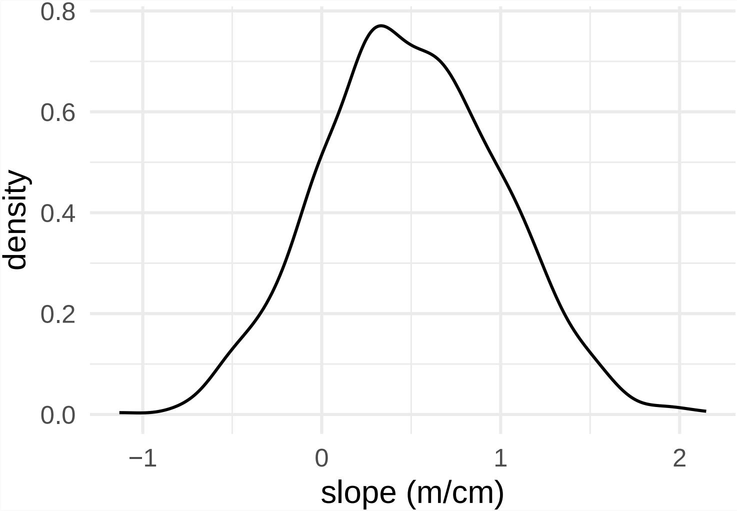
`sigma`: residual standard deviation (model error)

```r
priors <- c(
  set_prior('normal(30, 10)', class = 'Intercept'),
  set_prior('normal(0.5, 0.4)', class = 'b'),
  set_prior('normal(0, 5)', class = 'sigma')
)
```

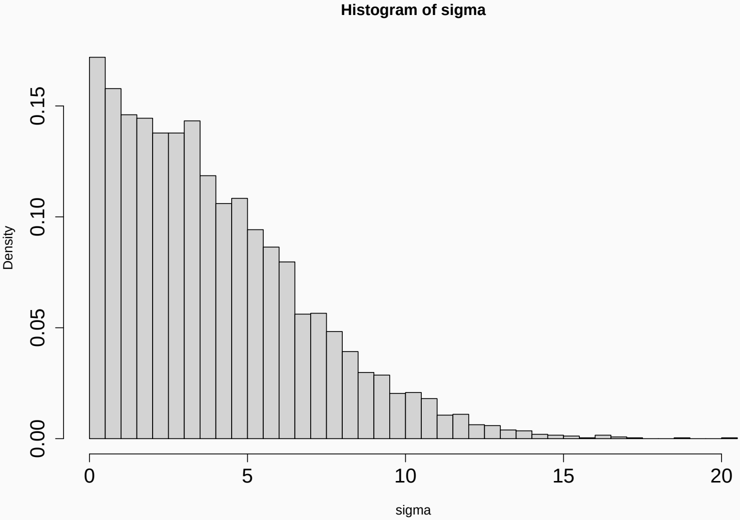
Prior for intercept (average height of 25-cm diameter tree)



Prior for slope



Prior for sigma (residual sd)

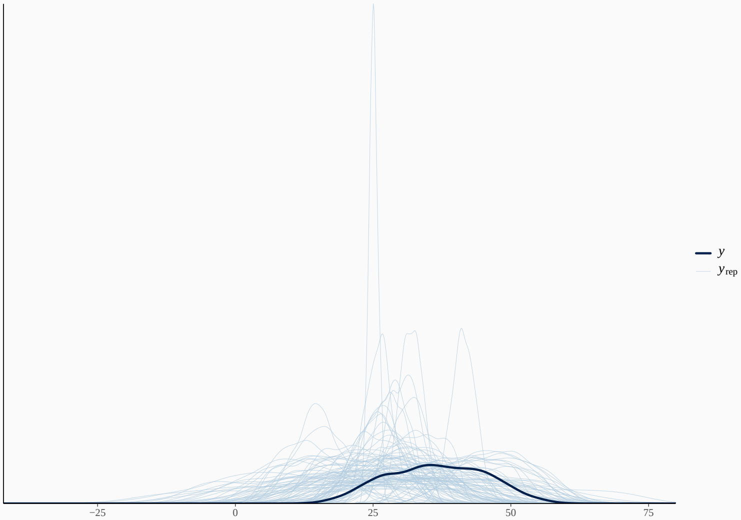


Prior predictive check

```
height.mod <- brm(height.formu,
         data = trees,
         prior = priors,
         sample_prior = 'only')
```

Prior predictive check

```r
pp_check(height.mod, ndraws = 100)
```



Fit model (now with data)

```r
height.mod <- brm(height.formu,
         data = trees,
         prior = priors)
```

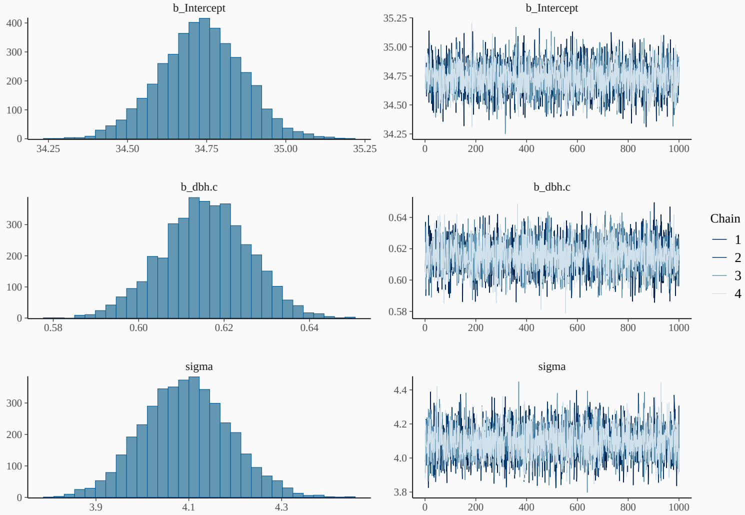
Model summary

```r
summary(height.mod)
```

```
 Family: gaussian
  Links: mu = identity
Formula: height ~ dbh.c
   Data: trees (Number of observations: 1000)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    34.73      0.13    34.47    34.99 1.00     4222     3247
dbh.c         0.62      0.01     0.60     0.64 1.00     4499     2673

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     4.09      0.09     3.92     4.28 1.00     4430     3153

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

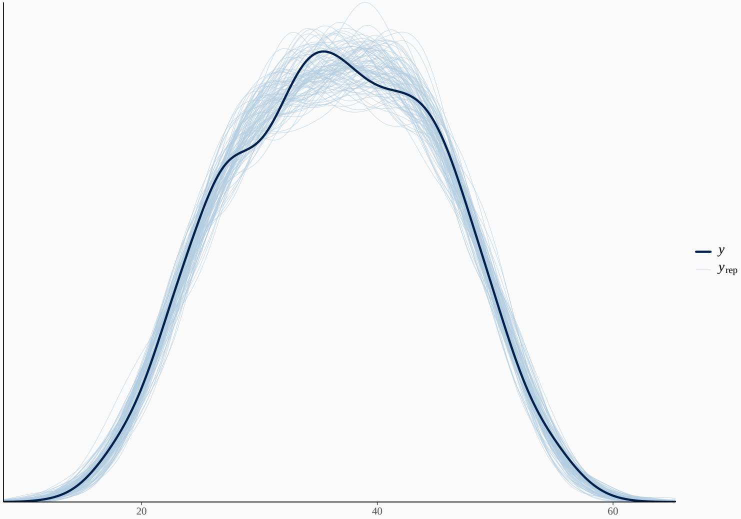
Model visualisation

```
plot(height.mod)
```



Posterior predictive checking

```r
pp_check(height.mod, ndraws = 100)
```



Interactive model exploration

```r
library('shinystan')
launch_shinystan(height.mod)
```

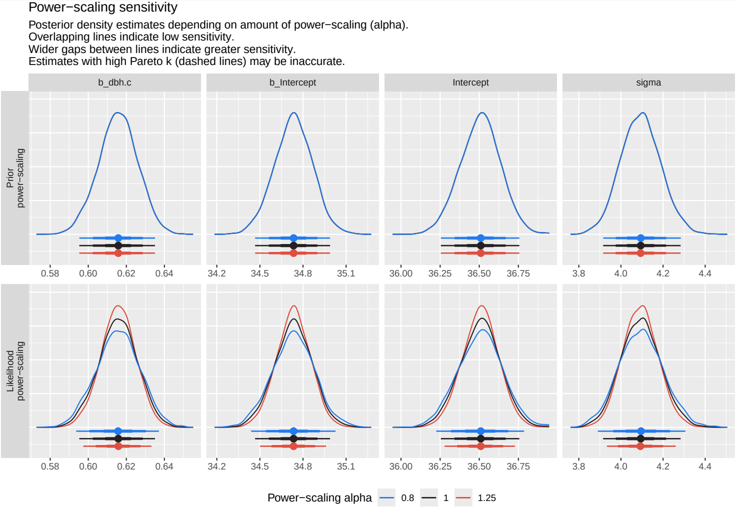
Prior sensitivity analysis

```r
library(priorsense)
powerscale_sensitivity(height.mod)
```

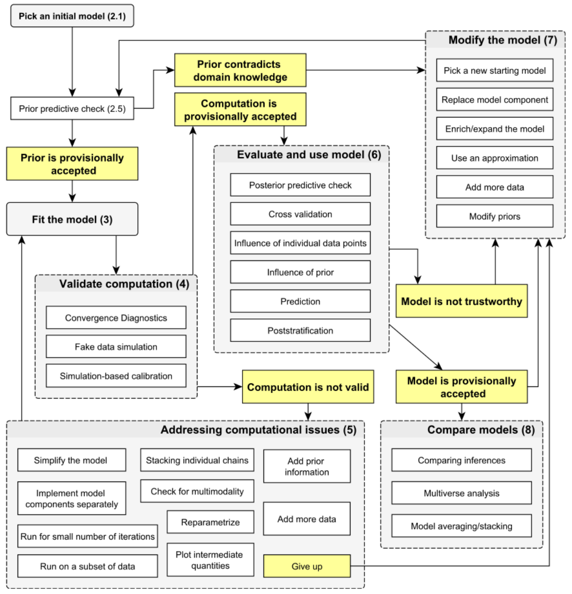
```
Sensitivity based on cjs_dist
Prior selection: all priors
Likelihood selection: all data

    variable prior likelihood diagnosis
 b_Intercept 0.001      0.084         -
     b_dbh.c 0.001      0.087         -
       sigma 0.003      0.100         -
   Intercept 0.001      0.085         -
```

Prior sensitivity analysis

```r
powerscale_plot_dens(height.mod)
```



The Bayesian workflow



https://arxiv.org/abs/2011.01808

Exercise

height ~ sex

To read more

Regression and other stories

Statistical Rethinking

Statistical rethinking with brms, ggplot2, and the tidyverse

Bayesian Population Analysis using WinBugs

Applied Hierarchical Modeling in Ecology

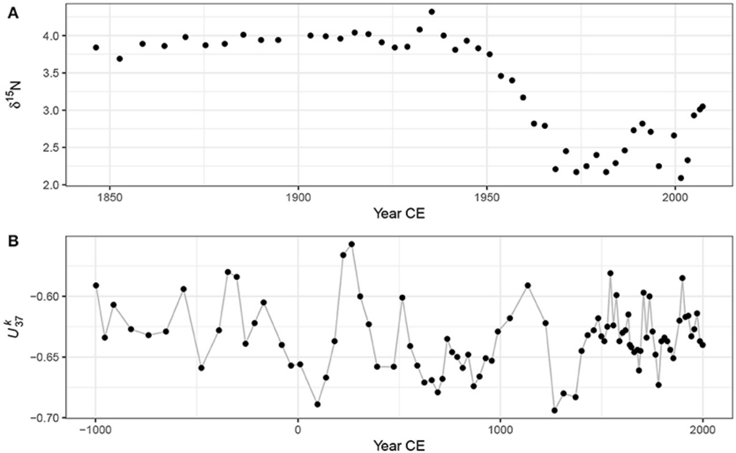
Stan user guide

Generalised Additive Models

Francisco Rodríguez-Sánchez

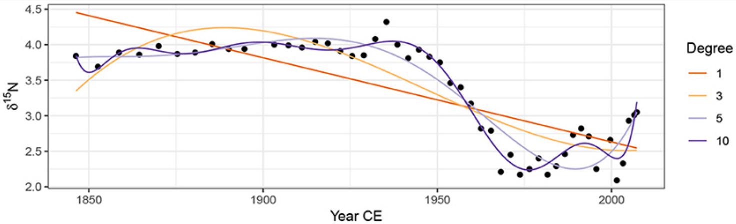
https://frodriguezsanchez.net

How do we model these time series?



Simpson 2018

How do we model these time series?



Simpson 2018

GAMs allow us to model non-linear relationships using smooths

Generalised Linear Model (GLM):

$$y = a + bx$$

Generalised Additive Model (GAM):

$$y = a + s(x)$$

Modelling non-linear time series with GAM

```r
isotopes <- readRDS('data/isotope.rds')
```

```
  Depth   d13C  TotalC d15N TotalN DryWeight     Year
1   0.2 -27.57  806.49 3.05  64.21       8.2 2007.254
2   0.4 -27.67  949.33 3.01  73.26       7.6 2006.510
3   0.8 -27.63 1305.52 2.93  93.25      11.6 2004.941
4   1.2 -27.62 1136.04 2.33  86.09       9.6 2003.269
5   1.6 -27.48 1028.27 2.09  93.80      10.9 2001.496
6   2.0 -27.39  809.91 2.66  79.98       9.9 1999.626
```

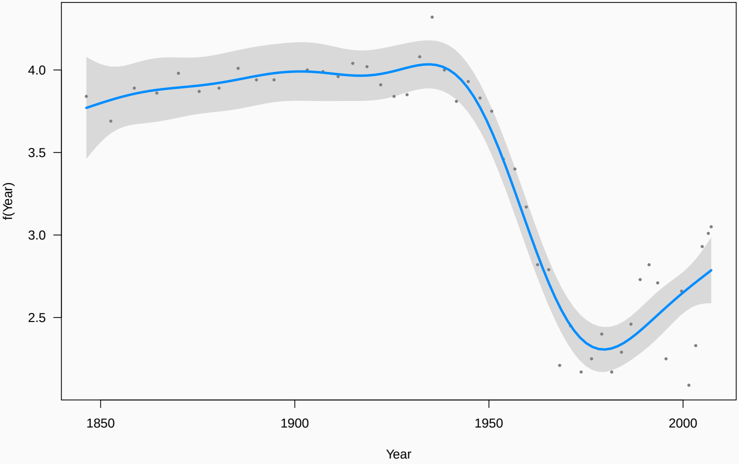
Modelling non-linear time series with GAM

```r
library('mgcv')
m <- gam(d15N ~ s(Year), data = isotopes, method = 'REML')
```

```

Family: gaussian
Link function: identity

Formula:
d15N ~ s(Year)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.30958    0.02805     118   <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(Year) 7.466  8.416 70.13  <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

R-sq.(adj) =  0.926   Deviance explained = 93.8%
-REML = 4.8282  Scale est. = 0.037771  n = 48
```

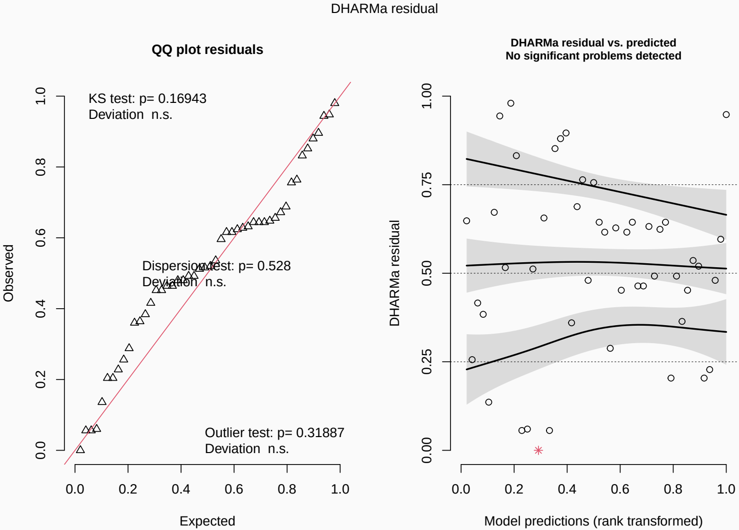
Visualising fitted GAM

```r
visreg(m)
```



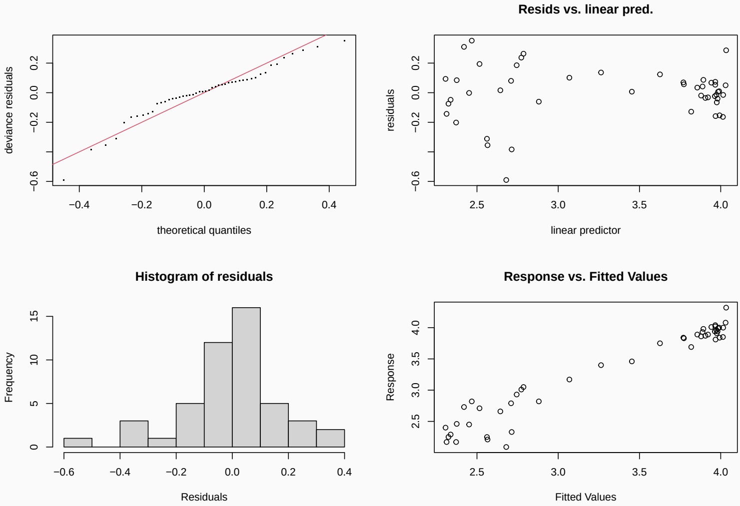
Checking fitted GAM

```
library('DHARMa')
simulateResiduals(m, plot = TRUE)
```



Checking fitted GAM

```r
gam.check(m)
```



Checking fitted GAM

```r
gam.check(m)
```

```
Method: REML   Optimizer: outer newton
full convergence after 6 iterations.
Gradient range [-3.747047e-11,2.175327e-11]
(score 4.828195 & scale 0.03777081).
Hessian positive definite, eigenvalue range [2.252432,23.50383].
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(Year) 9.00 7.47    0.73   0.025 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Increasing k

```
m <- gam(d15N ~ s(Year, k = 12), data = isotopes, method = 'REML')
gam.check(m)
```

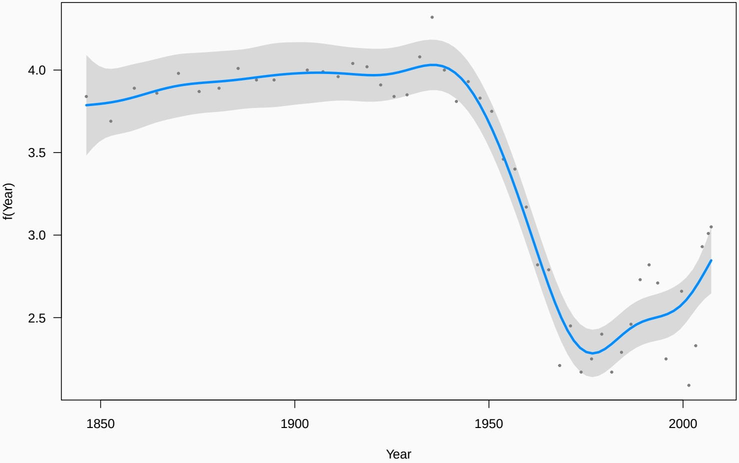
```
Method: REML   Optimizer: outer newton
full convergence after 5 iterations.
Gradient range [-1.437177e-06,2.684611e-07]
(score 4.742639 & scale 0.03687964).
Hessian positive definite, eigenvalue range [2.27223,23.61084].
Model rank =  12 / 12

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(Year) 11.00  8.12    0.76   0.045 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Increasing k

```
m <- gam(d15N ~ s(Year, k = 15), data = isotopes, method = 'REML')
gam.check(m)
```

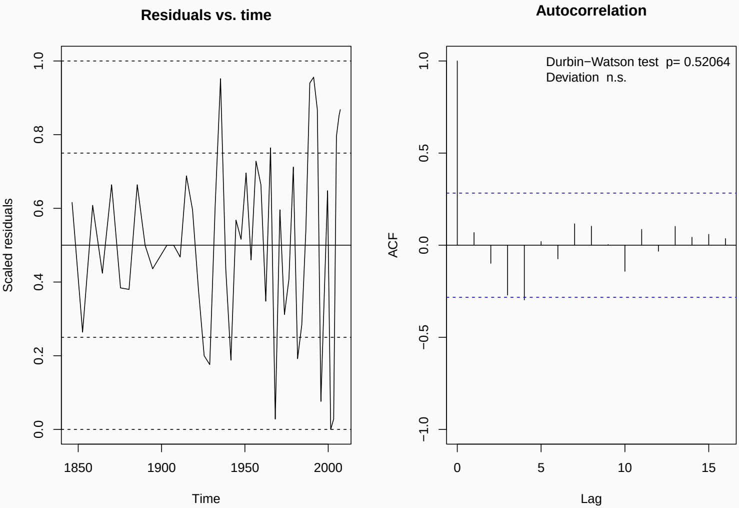
```
Method: REML   Optimizer: outer newton
full convergence after 5 iterations.
Gradient range [-2.100258e-10,1.486082e-10]
(score 3.973402 & scale 0.0329899).
Hessian positive definite, eigenvalue range [1.570402,23.80022].
Model rank =  15 / 15

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(Year) 14.00  9.28    0.85    0.11
```

Visualising fitted GAM



Checking fitted GAM

```
testTemporalAutocorrelation(simulateResiduals(m),
                            time = isotopes$Year)
```



Including temporal autocorrelation

```r
mod <- gamm(d15N ~ s(Year, k = 15), data = isotopes,
            correlation = corCAR1(form = ~ Year), method = 'REML')
```

```

Family: gaussian
Link function: identity

Formula:
d15N ~ s(Year, k = 15)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.30909    0.03489   94.84   <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(Year) 7.954  7.954 47.44  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.929
  Scale est. = 0.037268  n = 48
```

Modelling infant mortality

Modelling infant mortality

```r
mort <- read.csv('data/UN_GDP_infantmortality.csv')
```

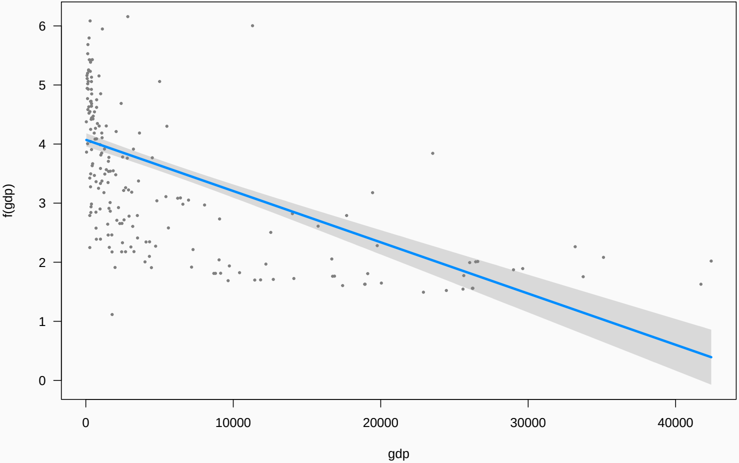
```
         country infant.mortality  gdp
1    Afghanistan              154 2848
2        Albania               32  863
3        Algeria               44 1531
4 American.Samoa               11   NA
5        Andorra               NA   NA
6         Angola              124  355
```

Modelling infant mortality with a GLM

```r
library('MASS')
mort.glm <- glm.nb(infant.mortality ~ gdp, data = mort)
```

```
Call:
glm.nb(formula = infant.mortality ~ gdp, data = mort, init.theta = 2.460991808,
    link = log)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.072e+00  5.727e-02   71.11   <2e-16 ***
gdp         -8.675e-05  6.221e-06  -13.95   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(2.461) family taken to be 1)

    Null deviance: 385.83  on 192  degrees of freedom
Residual deviance: 202.51  on 191  degrees of freedom
  (14 observations deleted due to missingness)
AIC: 1715
```

Modelling infant mortality with a GLM

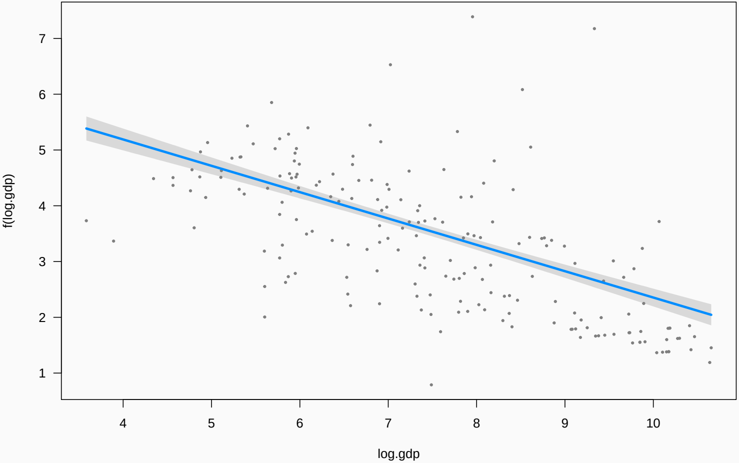


Modelling infant mortality with a GLM (log.gdp)

```r
mort$log.gdp <- log(mort$gdp)
mort.glm.log <- glm.nb(infant.mortality ~ log.gdp, data = mort)
```

```

Call:
glm.nb(formula = infant.mortality ~ log.gdp, data = mort, init.theta = 3.119314453,
    link = log)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  7.07818    0.20045   35.31   <2e-16 ***
log.gdp     -0.47238    0.02647  -17.85   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(3.1193) family taken to be 1)

    Null deviance: 478.54  on 192  degrees of freedom
Residual deviance: 198.03  on 191  degrees of freedom
  (14 observations deleted due to missingness)
AIC: 1667.7
```

Modelling infant mortality with a GLM (log.gdp)

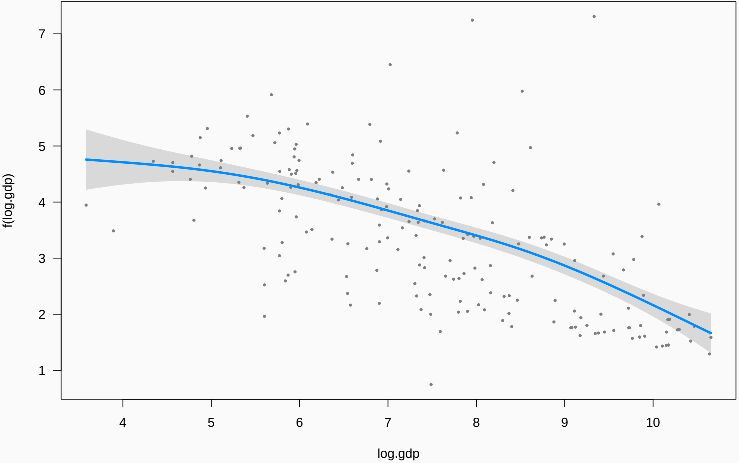


Modelling infant mortality with a GAM

```r
library('mgcv')
mort.gam <- gam(infant.mortality ~ s(log.gdp), family = nb, data = mort)
```

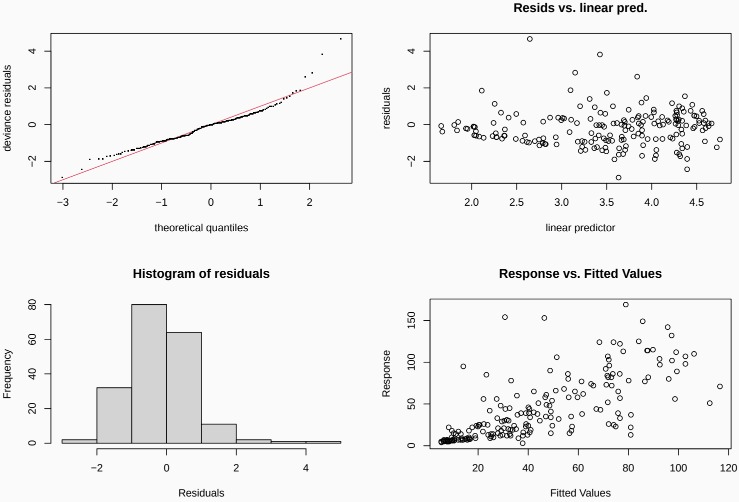
```
Family: Negative Binomial(3.251)
Link function: log

Formula:
infant.mortality ~ s(log.gdp)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.51137    0.04257   82.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
             edf Ref.df Chi.sq p-value
s(log.gdp) 3.134  3.937  329.9  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modelling infant mortality with a GAM



Checking GAM

```r
gam.check(mort.gam)
```



Checking GAM

```r
gam.check(mort.gam)
```

```
Method: REML   Optimizer: outer newton
full convergence after 4 iterations.
Gradient range [4.395199e-08,2.344699e-06]
(score 832.1006 & scale 1).
Hessian positive definite, eigenvalue range [0.4630879,82.57877].
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(log.gdp) 9.00 3.13    0.92    0.24
```

Comparing models

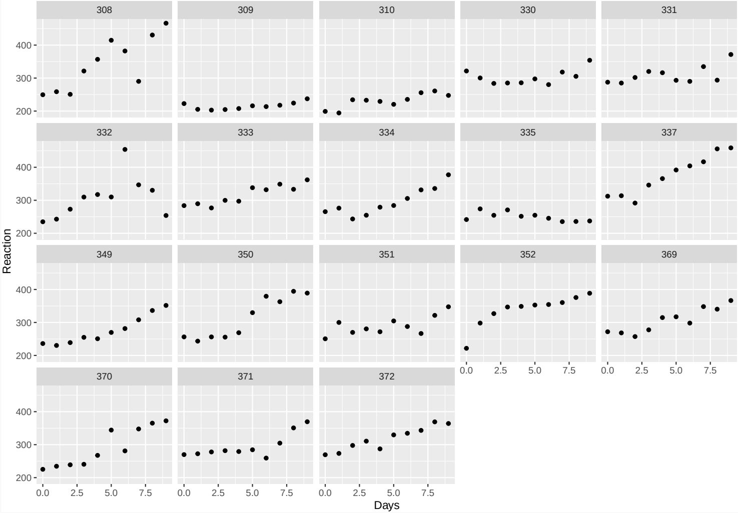
```r
library('easystats')
compare_performance(mort.glm, mort.glm.log, mort.gam)
```

```
# Comparison of Model Performance Indices

Name         |  Model |  AIC (weights) | AICc (weights) |  BIC (weights)
-----------------------------------------------------------------------
mort.glm     | negbin | 1715.0 (<.001) | 1715.1 (<.001) | 1724.7 (<.001)
mort.glm.log | negbin | 1667.7 (0.035) | 1667.9 (0.041) | 1677.5 (0.816)
mort.gam     |    gam | 1661.1 (0.965) | 1661.6 (0.959) | 1680.5 (0.184)

Name         |   RMSE | Sigma | Score_log | Score_spherical | Nagelkerke’s R2 |    R2
------------------------------------------------------------------------------------
mort.glm     | 31.089 |     1 |    -4.437 |           0.048 |           0.709 |      
mort.glm.log | 30.034 |     1 |    -4.356 |           0.053 |           0.836 |      
mort.gam     | 26.249 |     1 |    -4.296 |           0.049 |                 | 0.526
```

Generalised Additive Mixed Models (GAMM)

Reaction time with sleep deprivation

```r
library('lme4')
data('sleepstudy')
```



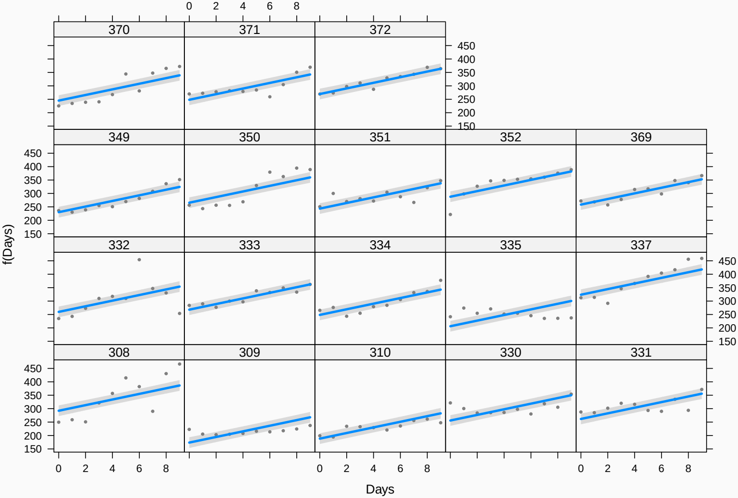
Modelling reaction time with sleep deprivation (GAMM)

```
sgamm <- gam(Reaction ~ s(Days) + s(Subject, bs = ”re”), data = sleepstudy, method = 'REML')
```

```
Family: gaussian
Link function: identity

Formula:
Reaction ~ s(Days) + s(Subject, bs = ”re”)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   298.51       9.05   32.98   <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(Days)     1.006  1.011 167.65  <2e-16 ***
s(Subject) 15.892 17.000  14.35  <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

R-sq.(adj) =  0.697   Deviance explained = 72.6%
-REML = 892.18  Scale est. = 960.43    n = 180
```

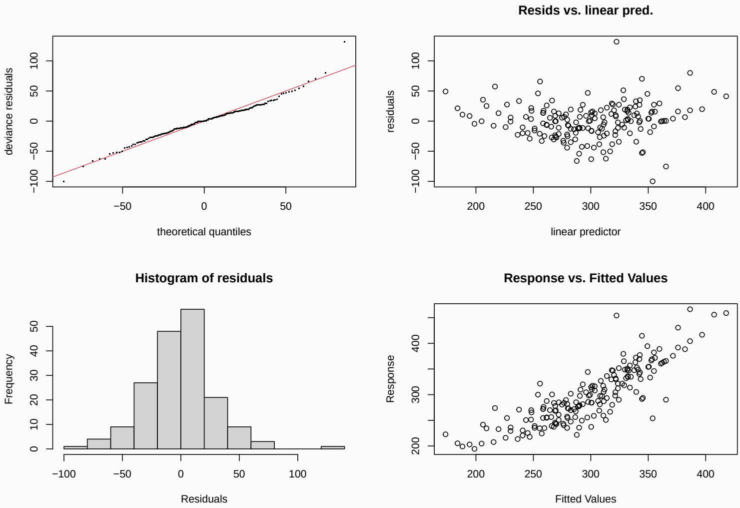
Modelling reaction time with sleep deprivation (GAMM)

```r
visreg(sgamm, xvar = 'Days', by = 'Subject')
```



Modelling reaction time with sleep deprivation (GAMM)

```
gam.check(sgamm)
```



Modelling reaction time with sleep deprivation (GAMM)

```r
gam.check(sgamm)
```

```

Method: REML   Optimizer: outer newton
full convergence after 7 iterations.
Gradient range [-0.0003899837,0.002002328]
(score 892.1778 & scale 960.4321).
Hessian positive definite, eigenvalue range [0.0004006477,89.76488].
Model rank =  28 / 28

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(Days)     9.00  1.01    1.07    0.81
s(Subject) 18.00 15.89      NA      NA
```

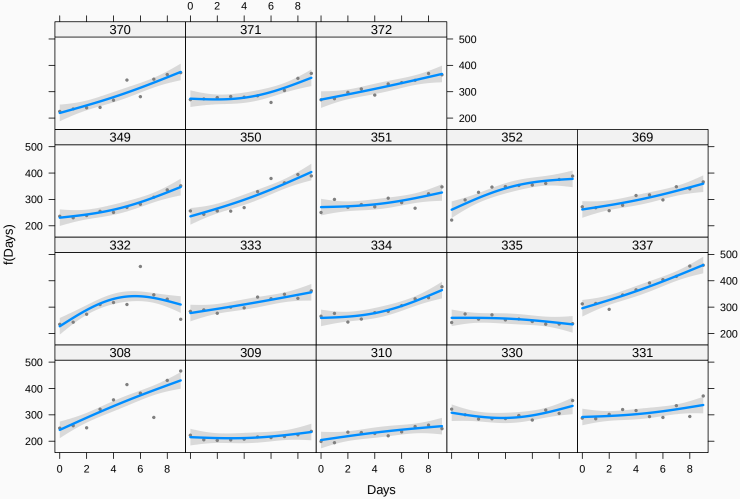
Modelling reaction time with sleep deprivation (GAMM)

```
sgamm <- gam(Reaction ~ s(Days, Subject, k = 3, bs = 'fs'),
                   data = sleepstudy, method = 'REML')
```

```
Family: gaussian
Link function: identity

Formula:
Reaction ~ s(Days, Subject, k = 3, bs = "fs")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   298.51       9.05   32.98   <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(Days,Subject) 45.67     53 17.11  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.835   Deviance explained = 87.7%
-REML = 883.64  Scale est. = 523.2     n = 180
```

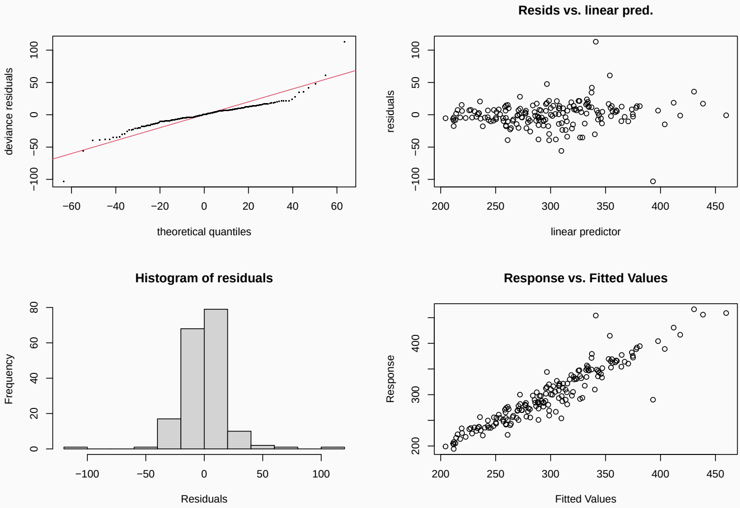
Modelling reaction time with sleep deprivation (GAMM)

```
visreg(sgamm, xvar = 'Days', by = 'Subject')
```



Modelling reaction time with sleep deprivation (GAMM)

```
gam.check(sgamm)
```



Modelling reaction time with sleep deprivation (GAMM)

```r
gam.check(sgamm)
```

```
Method: REML   Optimizer: outer newton
full convergence after 6 iterations.
Gradient range [-8.770542e-09,3.944987e-09]
(score 883.6432 & scale 523.1968).
Hessian positive definite, eigenvalue range [3.37416,91.60746].
Model rank =  55 / 55

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(Days,Subject) 54.0 45.7       1    0.47
```

To learn more

- Simon Wood's book

To learn more

- Simon Wood's book
- mgcv help

To learn more

- Simon Wood's book
- mgcv help
- G. Simpson's paper

To learn more

- Simon Wood's book
- mgcv help
- G. Simpson's paper
- G. Simpson's blog

To learn more

- Simon Wood's book
- mgcv help
- G. Simpson's paper
- G. Simpson's blog
- Noam Ross' course

To learn more

- Simon Wood’s book
- mgcv help
- G. Simpson’s paper
- G. Simpson’s blog
- Noam Ross’ course
- HGAM paper