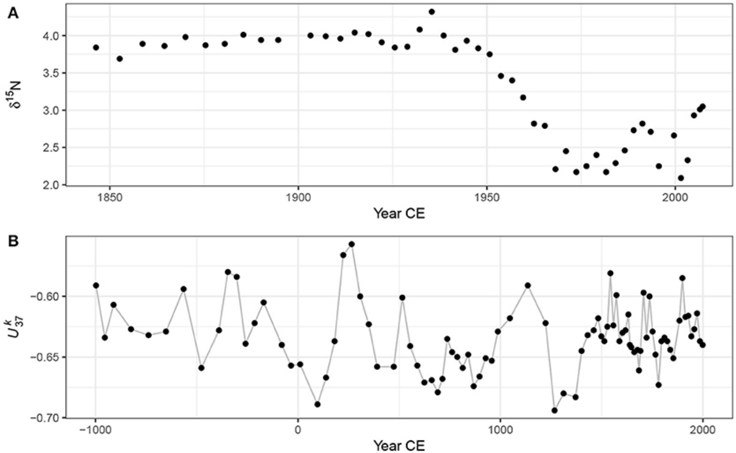
# Generalised Additive Models

---

Francisco Rodríguez-Sánchez

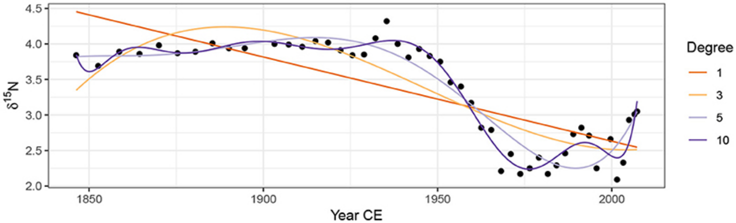
https://frodriguezsanchez.net

# 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
```

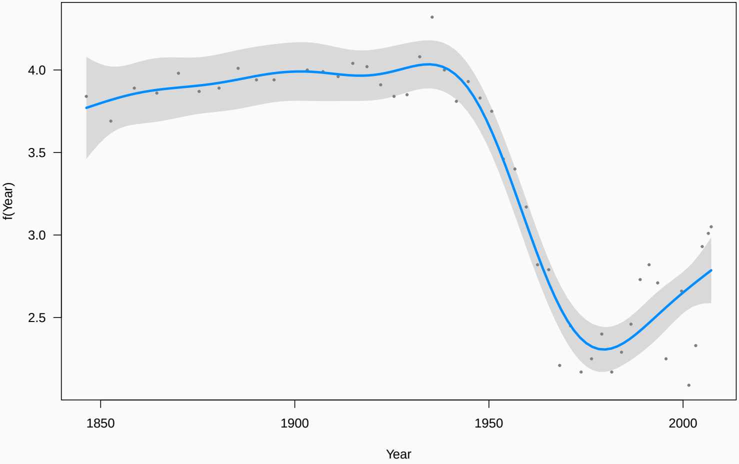
# Modelling non-linear time series with GAM

```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
```

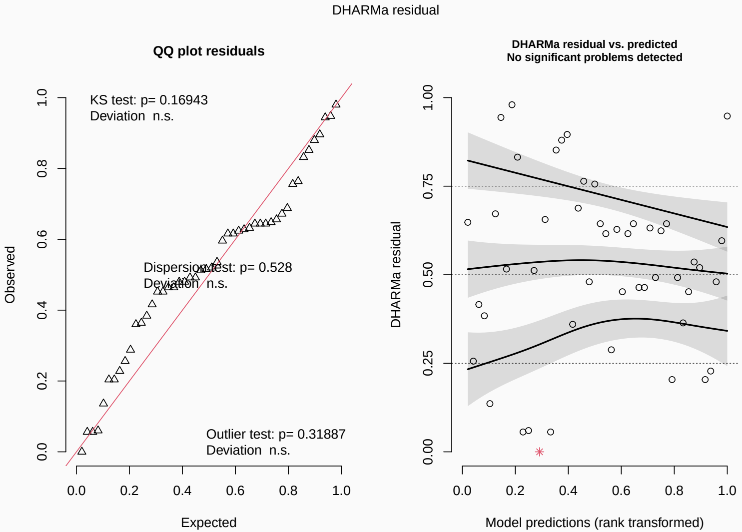
# Visualising fitted GAM

```
visreg(m)
```



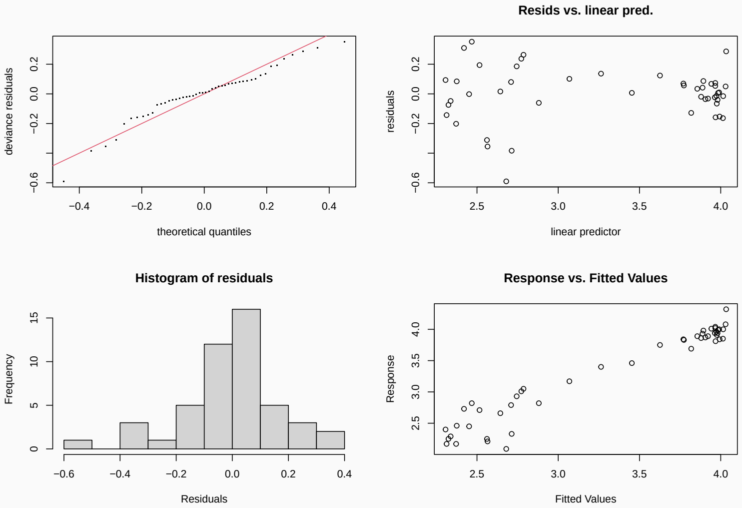
# Checking fitted GAM

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



# Checking fitted GAM

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



# Checking fitted GAM

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

```
Method: REML   Optimizer: outer newton
full convergence after 6 iterations.
Gradient range [-3.752376e-11,2.176748e-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    0.02 *
---
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)
```

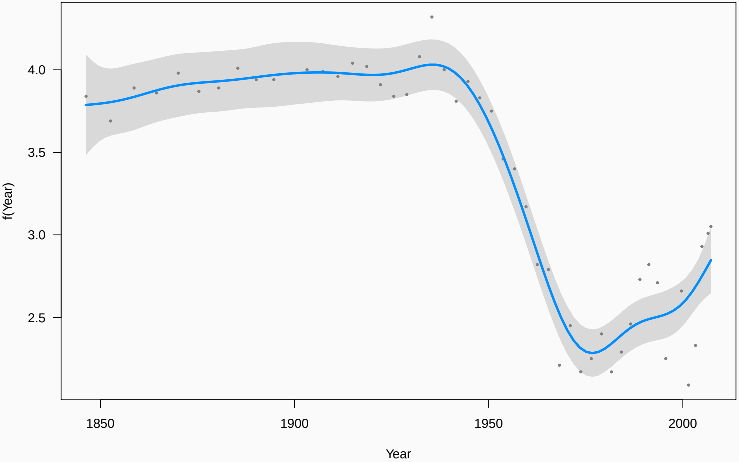
```
Method: REML   Optimizer: outer newton
full convergence after 5 iterations.
Gradient range [-1.437177e-06,2.684611e-07]
(score 4.742639 & scale 0.03687964).
Hessian positive definite, eigenvalue range [2.27223,23.61084].
Model rank =  12 / 12

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

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

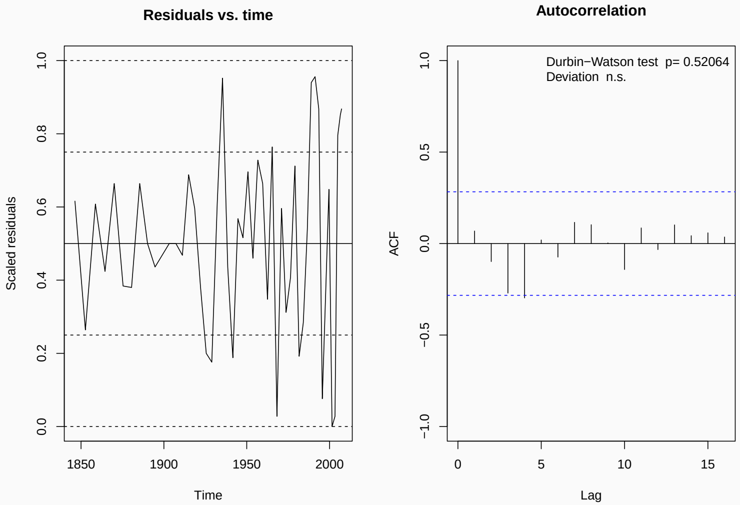
```
Method: REML   Optimizer: outer newton
full convergence after 5 iterations.
Gradient range [-2.100151e-10,1.486073e-10]
(score 3.973402 & scale 0.0329899).
Hessian positive definite, eigenvalue range [1.570402,23.80022].
Model rank =  15 / 15

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    0.09 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Visualising fitted GAM



# Checking fitted GAM

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



# Including temporal autocorrelation

```
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')
```

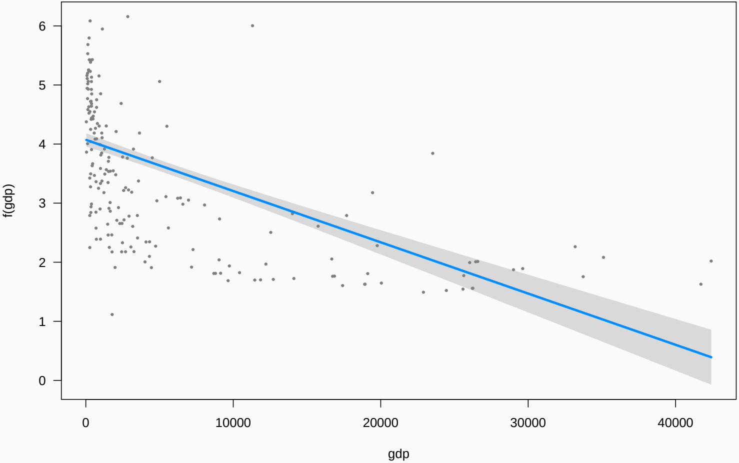
```
        country infant.mortality  gdp
1    Afghanistan              154 2848
2        Albania               32  863
3        Algeria               44 1531
4 American.Samoa               11   NA
5        Andorra               NA   NA
6         Angola              124  355
```

# 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

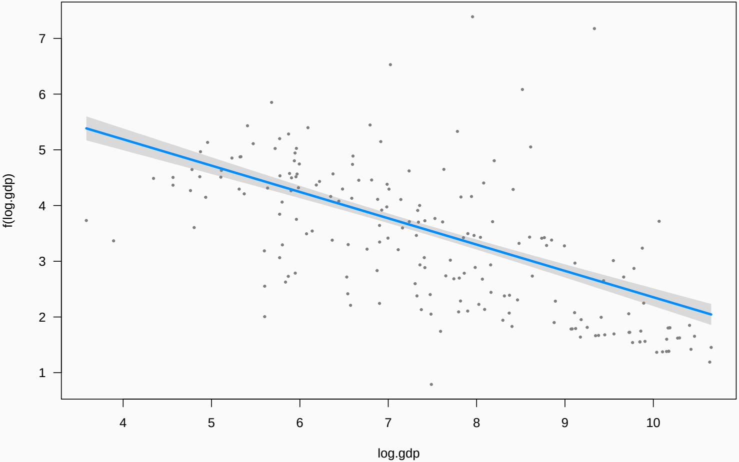


# Modelling infant mortality with a GLM (log.gdp)

```
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)

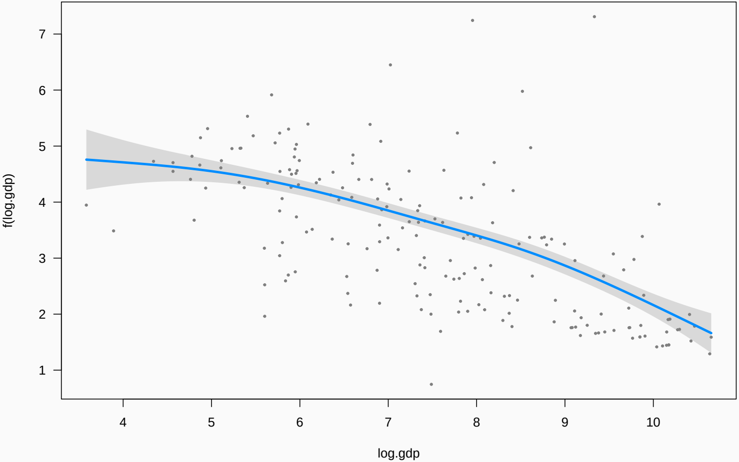


# Modelling infant mortality with a GAM

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

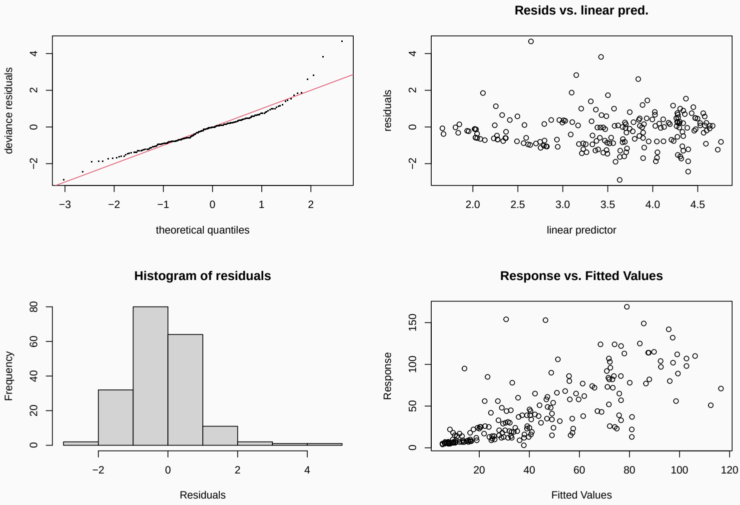
```

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

```
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

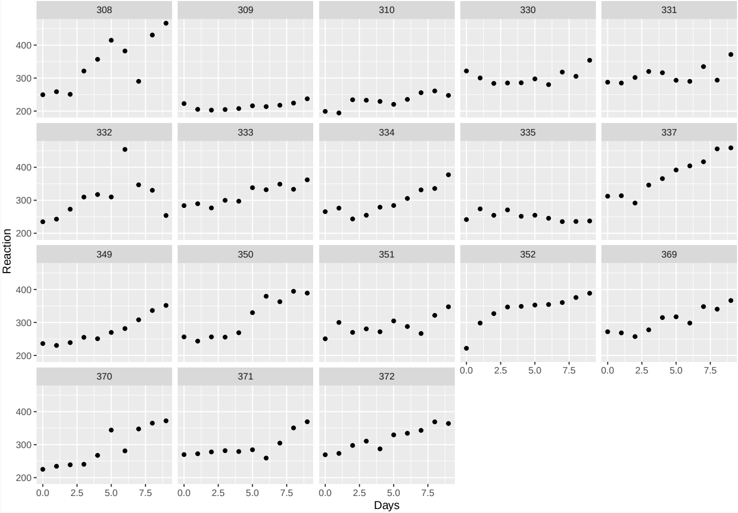
```r
library('easystats')
compare_performance(mort.glm, mort.glm.log, mort.gam)
```

```
# 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 | 1.000 |    -4.437 |           0.048 |           0.709 |
mort.glm.log | 30.034 | 1.000 |    -4.356 |           0.053 |           0.836 |
mort.gam     | 26.249 | 1.000 |    -4.296 |           0.049 |                 | 0.526
```

# Generalised Additive Mixed Models (GAMM)

# Reaction time with sleep deprivation

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



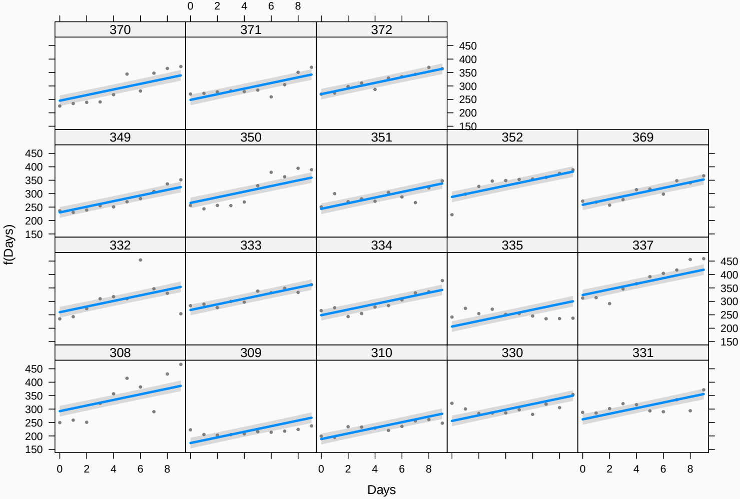
# Modelling reaction time with sleep deprivation (GAMM)

```
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
```

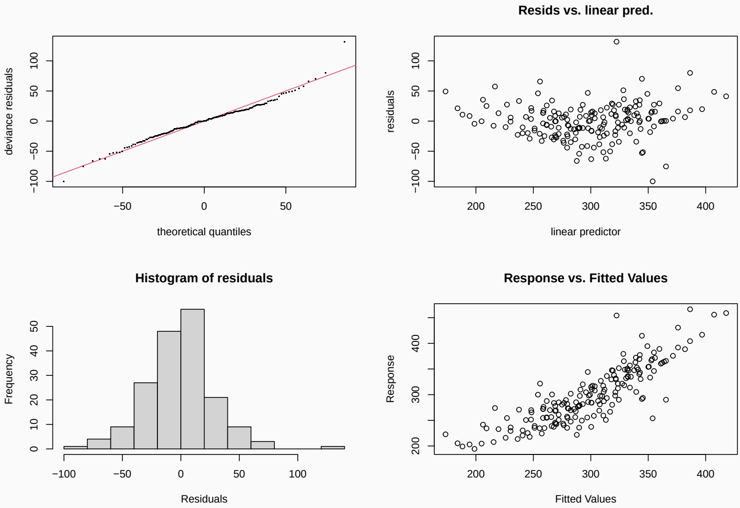
# Modelling reaction time with sleep deprivation (GAMM)

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



# Modelling reaction time with sleep deprivation (GAMM)

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



# Modelling reaction time with sleep deprivation (GAMM)

```
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    0.82
s(Subject) 18.00 15.89      NA      NA
```

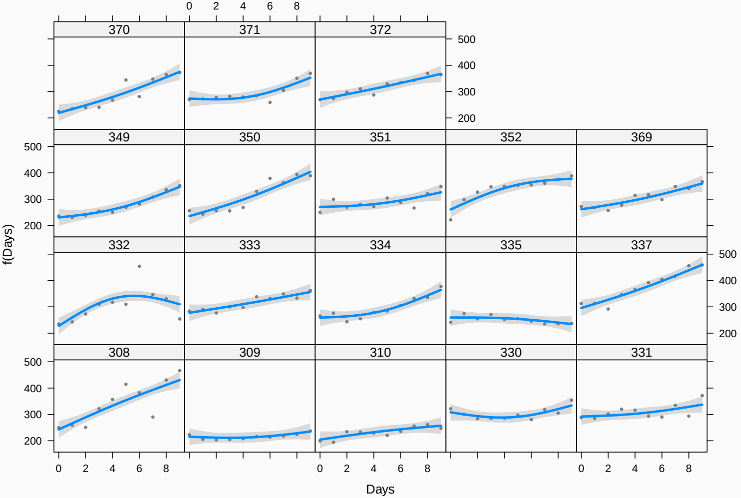
# Modelling reaction time with sleep deprivation (GAMM)

```
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
```

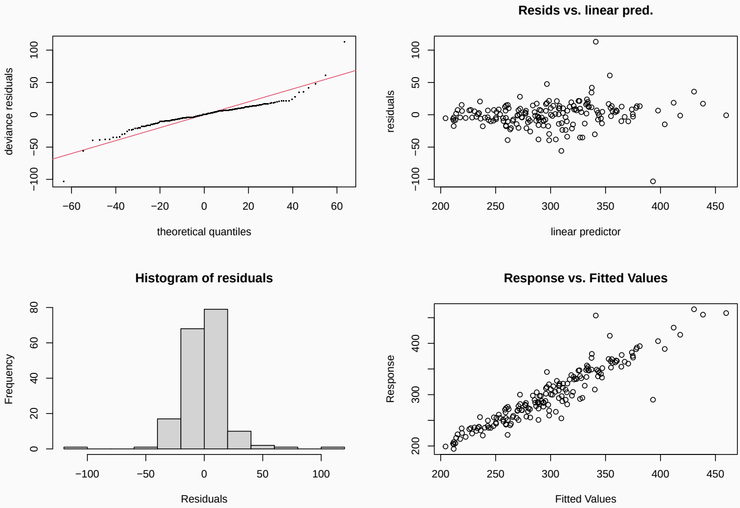
# Modelling reaction time with sleep deprivation (GAMM)

```
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 [-8.770485e-09,3.94499e-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    0.46
```

# 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