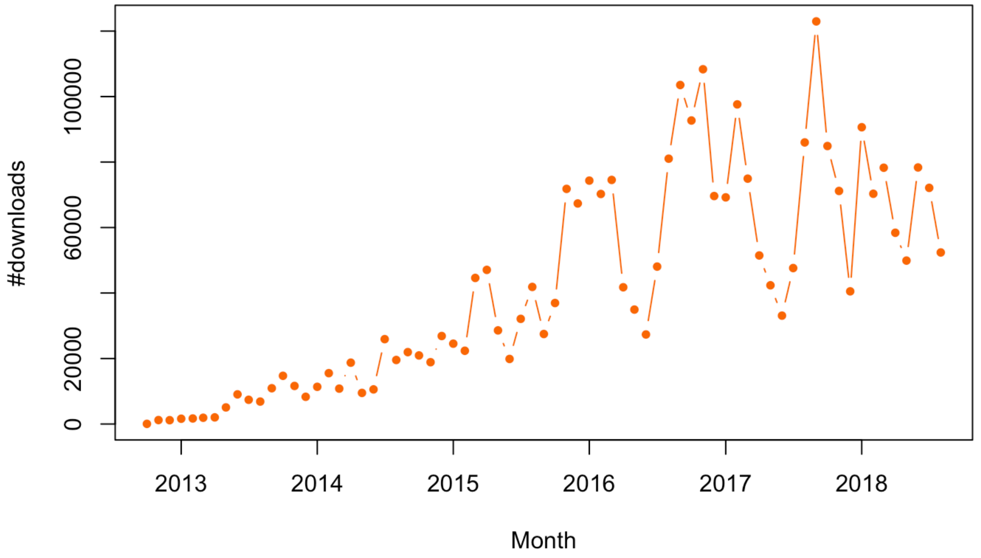
Generalized Additive Models: mgcv package

MGCV

mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation

Version:	1.8-24
Priority:	recommended
Depends:	R (≥ 2.14.0), nlme (≥ 3.1-64)
Imports:	methods, stats, graphics, Matrix
Suggests:	splines, parallel, survival, MASS
Published:	2018-06-18
Author:	Simon Wood
Maintainer:	Simon Wood <simon.wood at r-project.org>
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]



Included in the R base packages

MGCV

The principal functions we will use are `gam` or `bam`

```
library(mgcv)

gam(y ~ x0 + s(x1)+s(x2)+s(x3,x4), data=Data, family=gaussian())
```

- the model is enter via the `formula` syntax
- non-linear effects are entered with the syntax `s`
- bivariate effects can be either enterd with the syntax `s` or `te` if the smoothness is different between axes

MGCV

The function `s` has different arguments:

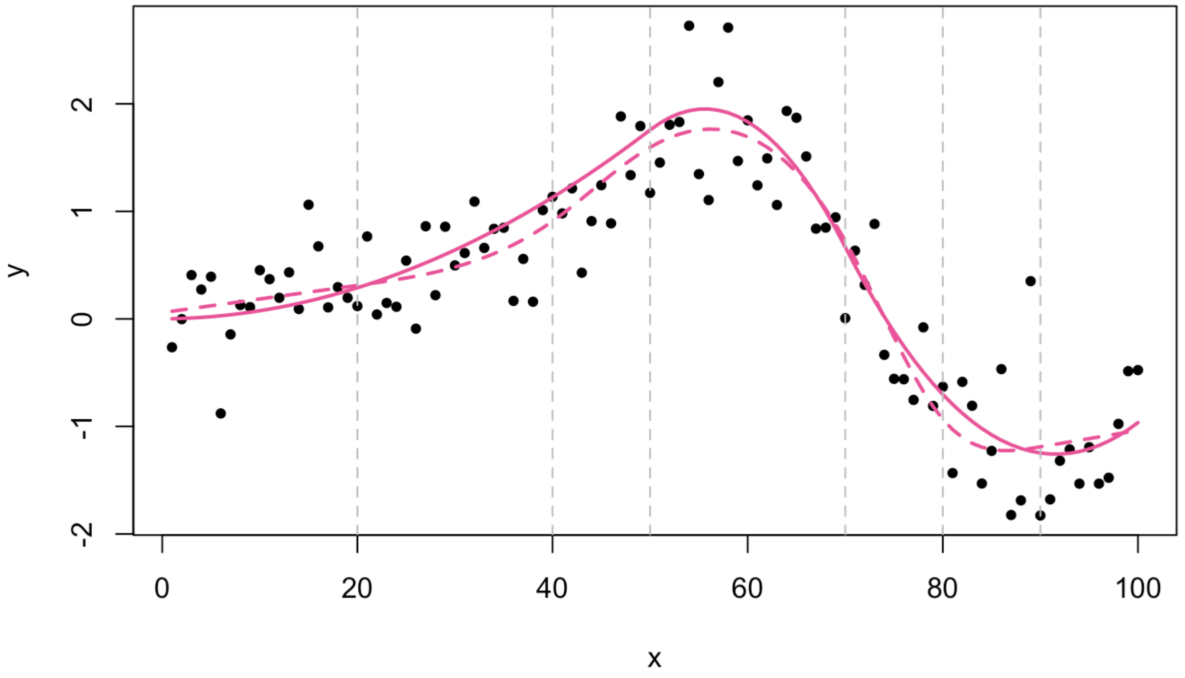
- `k` : the dimension of the basis used to represent the smooth term
- `bs` : indicating the (penalized) smoothing basis to use, could be:
    - `bs="tp"` , thin plate regression splines
    - `bs="ds"` , Duchon splines (generalize the thin plate slines)
    - `bs="cr"`, cubic regression splines
    - `bs="cc"`, cyclic cubic regression splines
    - `bs="ps"`, P-splines as proposed by Eilers and Marx (1996), they combine a B-spline basis, with a discrete penalty on the basis coefficients
    - `bs="ad"` univariate and bivariate adaptive smooths are available (see adaptive.smooth).

MGCV

It is also possible to set the position of knots:

```
knot <- c(20,40,50,70, 80, 90)
g <- gam(y ~ s(x, k=6, bs='cr'), knots=list(x=knot), sp=0)
```

otherwise knots are positionned on a regular partition of the quantiles



MGCV

Model summary

```r
g <- gam(y ~ s(x, k=10, bs='cr')+s(z, k=10, bs='cr'))
summary(g)
```

Linear terms

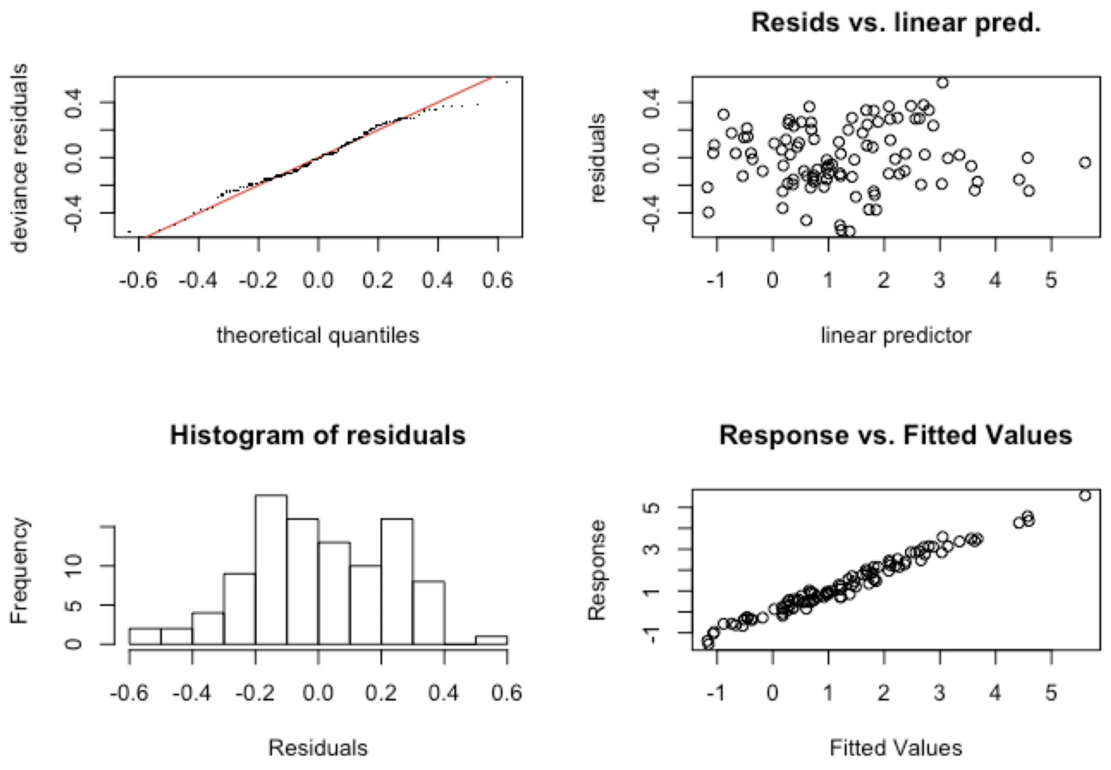
Smooth terms

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(x, k = 10, bs = "cr") + s(z, k = 10, bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.18684    0.02436   48.71   <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) 7.254  8.258 204.3  <2e-16 ***
## s(z) 8.555  8.920 178.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.97   Deviance explained = 97.5%
## GCV = 0.071359  Scale est. = 0.059364  n = 100
```

MGCV

Model check

```
g <- gam(y ~ s(x, k=10, bs='cr')+s(z, k=10, bs='cr'))
gam.check(g)
```



MGCV

Choice of k

in practice k-1 (or k) sets the upper limit on the degrees of freedom associated with an s smooth (1 degree of freedom is usually lost to the identifiability constraint on the smooth)

- k should be chosen large enough to capture the function complexity
- k should be chosen small enough to avaoid overfitting and maintain reasonable computational efficiency

'large' and 'small' are dependent on the particular problem being addressed.

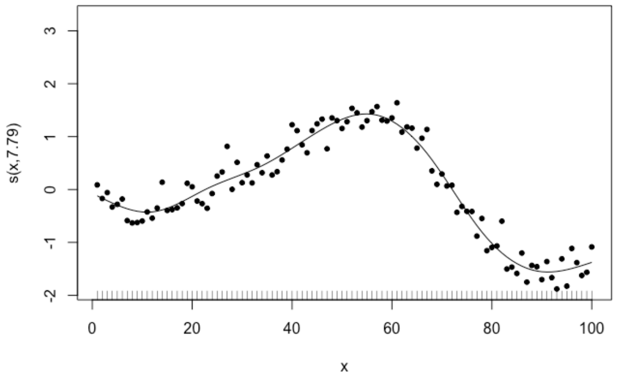
A useful general purpose approach goes as follows (see `mgcv::choose.k`)

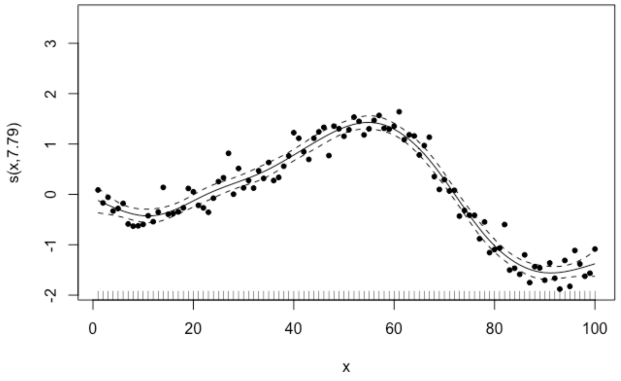
- fit your model and extract the deviance residuals
- for each smooth term in your model, fit an equivalent, single, smooth to the residuals, using a substantially increased k to see if there is pattern in the residuals that could potentially be explained by increasing k.

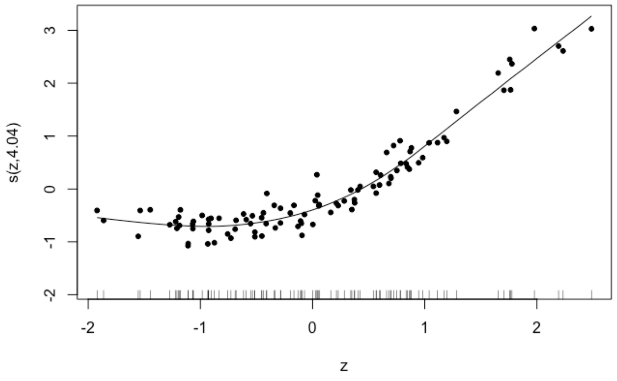
The obvious, but more costly, alternative is simply to increase the suspect k and refit the original model. If there are no statistically important changes as a result of doing this, then k was large enough.

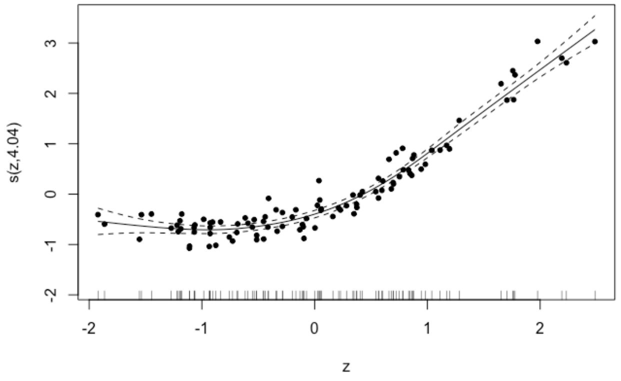
MGCV

```r
g <- gam(y ~ s(x, k=10, bs='cr')+s(z, k=10, bs='cr'))
plot(g, residuals=T,rug=T, se=F, pch=20)
```





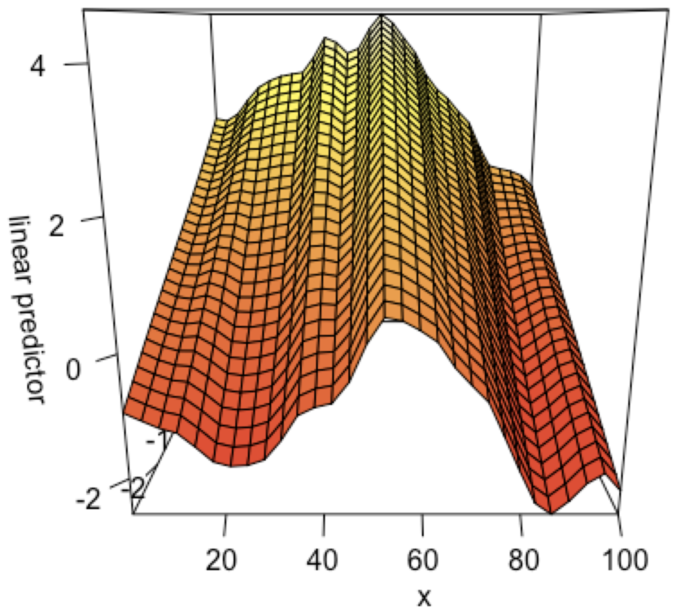




MGCV

```
g <- gam(y ~ s(x,z))

vis.gam(g,view=c("x","z"),plot.type="persp",box=T
,ticktype="detailed")
```



MGCV

forecasting

```r
g <- gam(y ~ s(x,z), data=data0)
g.forecast <- predict(g, newdata=data1)
```