

mgcv tips sheet

What is mgcv ?

`mgcv` is an R package for fitting generalized additive models (GAMs). That means we can fit models where the predictors are smooth functions of the covariates. Often these smooth functions are splines, but that's not all they can be.

The main functions in mgcv

`gam`
For fitting GAMs

`gamm`
For fitting generalized additive mixed models. Can include correlation structures and performance can be better for random effects. You can specify random effects using `lme` syntax.

`bam`
For fitting big additive models. Includes some special tricks for fitting to large datasets.

Formula

formula=

We can write a model formula in `mgcv` just as we can when we use `lm` or `glm`, with some additions.

`s()` is the general setup for a smooth.

`te()` interaction via tensor product.

Distributions

	family=
Binomial	<code>binomial</code>
Normal	<code>gaussian</code>
Gamma	<code>Gamma</code>
Inverse normal	<code>inverse.gaussian</code>
Poisson	<code>poisson</code>
Quasi	<code>quasi</code>
Quasi-binomial	<code>quasibinomial</code>
Quasi-Poisson	<code>quasipoisson</code>
Tweedie	<code>tw/Tweedie</code>
Negative binomial	<code>nb/negbin</code>
Beta	<code>betar</code>
Censored normal	<code>cnorm</code>
Ordered categorical	<code>ocat</code>
Scaled t	<code>scat</code>
Zero inflated Poisson	<code>zip</code>
Zero inflated Poisson location-scale	<code>ziplss</code>
Cox proportional hazards	<code>cox.ph</code>
Generalized extreme value location-scale	<code>gevlss</code>
Normal	<code>gaulss</code>
location-scale model	<code>mvn</code>
Multivariate normal	<code>gammals</code>
Gamma	<code>gumbls</code>
location-scale	<code>gumbls</code>
Gumbel	<code>gumbls</code>
location-scale	<code>gumbls</code>
Multinomial	<code>multinom</code>
Tweedie	<code>twlss</code>
location-scale	<code>twlss</code>
Sinh-arcsinh	<code>shash</code>
location-scale-shape	<code>shash</code>
General family	<code>gfam</code>

Smoothers

Using the `bs=` argument in `s()`, `te()`, etc. Further details can be found in `?smooth.construct.*.smooth.spec`

Univariate only smoothers

Cubic regression splines `cr`

Cubic regression splines with shrinkage `cs`

Cyclic cubic splines `cc`

B-splines `bs`

P-splines `ps`

Special smoothers

Adaptive smoothers `ad`

Factor-smooth interactions `sz`

Random factor-smooth interactions `fs`

Smoothers in ≥ 1 dimension

Thin plate regression splines `tp`

Thin plate regression splines within shrinkage `ts`

Duchon splines `ds`

Random effects `re`

Markov random fields `mrf`

Gaussian process smooths `gp`

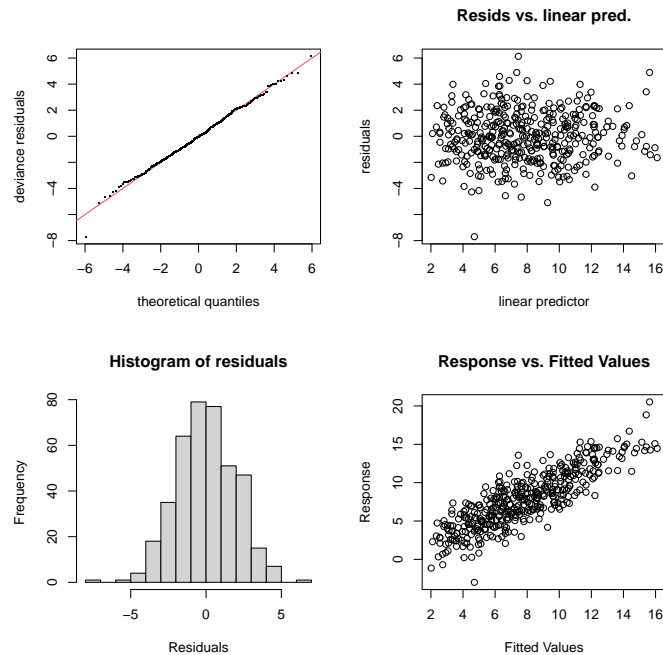
Smoothers in 2 dimensions

Splines on the sphere `sos`

Soap film smoothing `so` (`sw` and `sf`)

Model checking

gam.check



Top left: Quantile-quantile plot: points should be close to the line, meaning residuals are normally distributed.

Bottom left: Histogram of residuals: again, looking for normal(ish) distribution.

Top right: Residuals vs. linear predictor: looking for no increase or decrease in spread with increasing linear predictor value, otherwise we have heteroskedasticity.

Bottom right: Response vs. linear predictor, expecting tight line relationship indicating good agreement between the predictions and data.

Knots and basis complexity

General strategy: check k and double if too small.
When do we know k is too small?

Example

```
> gam.check(b)

Method: GCV   Optimizer: magic
Smoothing parameter selection converged
after 12 iterations.
The RMS GCV score gradient at convergence
was 1.739918e-07 .
The Hessian was positive definite.
Model rank = 37 / 37
```

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(x0)	9.0	2.5	1.04	0.85
s(x1)	9.0	2.4	1.03	0.69
s(x2)	9.0	7.7	0.97	0.28
s(x3)	9.0	1.0	1.03	0.68

Just as it says, check the p-value and k -index columns!
Double k if necessary.

Fitting criterion method=

"GCV.Cp"	Generalized cross validation, default
"REML"	REstricted Maximum Likelihood, preferred
"ML"	Maximum Likelihood
"NCV"	Neighbourhood Cross-Validation

Extras

gam.mh	Metropolis-Hastings sampling of the posterior
concurvity	Assess concavity between terms
gam.vcomp	Random effects style output
gamSim	Simulate GAM-type data
inSide/in.out	point-in-polygon test
jagam	Generate JAGS/Nimble code
new.name	Generate a variable name
place.knots	Place knots evenly
rmvn	Generate multivariate normal deviates

Extra help

?gam.models	Fitting fancy models
?linear.functionals	How to use by=
?random.effects	Random effects syntax
?mgcv.FAQ	frequently asked questions
?mgcv.parallel	Info on parallelisation
?missing.data	What to do about missing data
?choose.k	How to select basis size
?one.se.rule	Making smoother smooth models

Other packages

scam	Shape constrained smoothing
gratia	Plotting with ggplot2
mgcViz	Fancy plotting
qgam	Quantile GAMs
gamm4	Random effects based on lme4

Useful references

Wood. Generalized Additive Models. An Introduction with R. 2nd ed. CRC Press, 2017

Pedersen, Miller, Simpson and Ross. Hierarchical Generalized Additive Models in Ecology: An Introduction with mgcv. PeerJ (2019). <https://doi.org/10.7717/peerj.6876>