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. (See options `discrete`, `cluster` and `nthreads`.)

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

`ti()` interaction via tensor product (ANOVA decomposition)

`t2()` interaction via tensor product (via identity penalty matrices)

Distributions `family=`

Continuous

Normal	`gaussian`
Multivariate normal	`mvn`
Normal location-scale	`gaulss`
Scaled ***t***	`scat`
Inverse normal	`inverse.gaussian`
Quasi	`quasi`

Discrete

Binomial	`binomial`
Quasi-binomial	`quasibinomial`
Ordered categorical	`ocat`
Multinomial	`multinom`

Count

Poisson	`poisson`
Quasi-Poisson	`quasipoisson`
Tweedie	`tw` or `Tweedie`
Tweedie location-scale	`twlss`
Negative binomial	`nb` or `negbin`
Zero inflated Poisson	`ziP`
Zero inflated Poisson location-scale	`ziplss`

Range constrained

Beta	`betar`
Gamma	`Gamma`
Gamma location-scale	`gammals`
Censored normal	`cnorm`

Extremes

Generalized extreme value location-scale	`gevlss`
Gumbel location-scale	`gumbls`

Misc

Cox proportional hazards	`cox.ph`
Sinh-arcsinh location-scale-shape	`shash`
General family	`gfam`

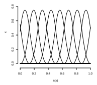
Smoothers

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`



B-splines `bs`

P-splines `ps`

Special smoothers

Cyclic cubic splines `cc`

Adaptive smoothers `ad`

Factor-smooth interactions `sz`

Random factor-smooth interactions `fs`

Smoothers in $\geq$ 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`

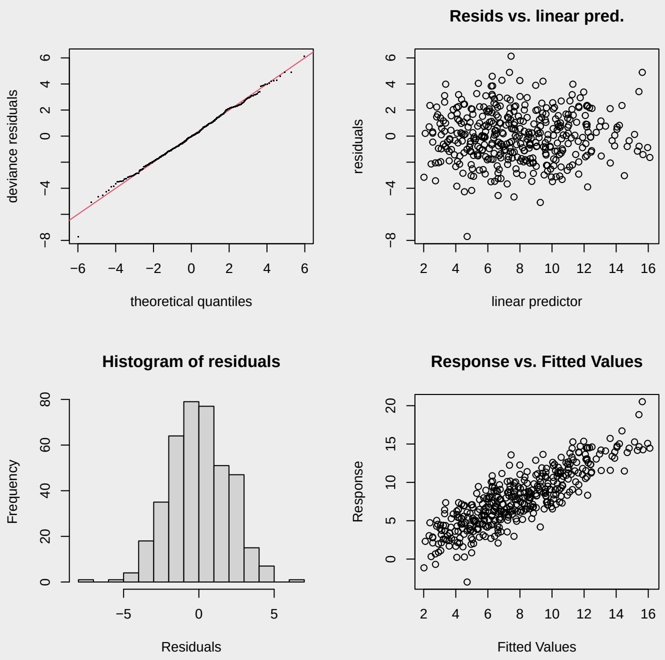
Soap film smoothing `so` (`sw` and `sf`)

`mgcv` is developed by Simon Wood • CC BY David L Miller • `dave.miller@bioss.ac.uk`

`https://github.com/dill/mgcv-tipsheet` • Tips sheet version 1 • Updated: 11/2023

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

```
> k.check(b)
      k'       edf   k-index p-value
s(x0)  9 2.500168 1.0448166  0.8000
s(x1)  9 2.401079 1.0267384  0.7200
s(x2)  9 7.697714 0.9691083  0.2300
s(x3)  9 1.000000 1.0297794  0.7325
```

Check the `p-value`/`k-index` columns, if $<<$ 1 double `k`.

Prediction

predict

`unconditional=TRUE` the smoothing parameter uncertainty

`type=` argument changes the type of prediction

default	on the link scale
`"response"`	to put on the response scale
`"iterms"`	to give per term predictions
`"lpmatrix"`	for a prediction matrix

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 concurvity 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 besed 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