First session: computer lab exercises

Assuming you have installed all the relevant software, you should get all the material for the workshop from https://github.com/mfasiolo/workshop_ENBIS18. Download it as a zip file and extract it. Set the working directory to the "exercises" subfolder, then you should be able to load any of the datasets by doing:

```
load(file = "data/AnyOfTheDataSets.rda")
```

In this session you could try one or more of the following exercises on electricity demand forecasting:

1. Forecasting electricity demand on GEFCom2014 data (solution in: "gefcom_small.html"). Simple exercise, featuring only 1D smooth effects on a relatively small data set.

2. Big GAM modelling of GEFCom14 electricity demand data (sol: "gefcom_big.html"). Featuring Big Data GAM methods and 2D tensor interactions.

3. Individual electricity demand modelling (solution in: "Ind_elect.html"). Featuring Big Data GAM methods and by-costumer smooth effects.

Otherwise you could try one of these other exercises, not focused on the electricity industry:

4. Mackerel egg data (sol: "Mackerel.html"). Featuring 2D spatial interactions.

5. Bone mineral density modelling (sol: "bone_density.html"). Featuring simple random effects.

6. Retinopathy among diabetics (sol: "Retinopath.html"). Features 2D smooth interactions and automatic variable selection.

7. $C0_2$ modelling (sol: "CO2_modelling.html"). Featuring cyclic seasonal smooths and the dangers of extrapolation.

8. Ozone modelling (sol: "Ozone.html"). Easy exercise, focusing on manual variable selection via p-values and residual checking.

9. Larynx cancer in Germany (sol: "Larinx.html"). Focused on spatial modelling using Markov Random Field effects.

but feel free to try `mgcv` and `mgcViz` on your own data.

Questions 4, 6, 7 and 9 have been adapted from Simon Wood's notes.

1 Forecasting electricity demand on GEFCom2014 data

Here we consider the electricity demand dataset taken from the GEFCom2014 challenge. The dataset covers the period January 2005 to December 2011 and it contains the following variables:

- `NetDemand` net electricity demand between 11am and 12am.
- `wM` instantaneous temperature.
- `wM_s95` exponential smooth of `wM`, that is `wM_s95[i] = a*wM[i] + (1-a)*wM_s95[i]` with `a=0.95`.
- `Posan` periodic index in `[0, 1]` indicating the position along the year.

- `Dow` factor variable indicating the day of the week (I think that 0=Sunday and 6=Saturday, but I am not sure).
- `Trend` progressive counter, useful for defining the long term trend.
- `NetDemand.24` lagged version of `NetDemand`, that is demand at the same time of the previous day.
- `Year` should be obvious.

Questions:

1. Load `mgcViz` and the data (`load("data/gefcom_small.rda")`). Use `gamV` to fit a Gaussian GAM where the model formula (e.g. `y~s(x)`) contains: smooth effects for `wM`, `wM_s95`, `Posan` (optionally use a cyclic basis for the latter by doing `s(Posan, bs="cc")`); parametric effects for `Trend`, `Dow` and `NetDemand.24`. In the call to `gamV` set the argument `aViz=list(nsim = 50)` to have some simulated responses for residuals checking. Plot all the fitted effects using `plot`.

2. Use `check1D` with the `l_gridCheck1D` layer to check that the mean of the residuals does not depart too much from 0, depending on the value of `Trend`. Do you see any systematic pattern? Use the function `check` to see whether you should increase `k` for any of the smooth effects.

3. Refit the model but now include a smooth effect for `Trend` (with `k=6`) and increase the basis dimension of the effects of `wM`, `wM_s95`, `Posan` to `k =` 20, 15 and 15 respectively. Compare this model to the old one in terms of `AIC`, re-check the residuals using `check1D` and recheck the bases dimension using `check`. Does everything look good? Increasing too much the basis dimension for `Trend` is not a good idea, why?

4. Use `qq` to produce a QQ-plot of the residuals, do you see any problem? Refit the same model, but now use a scaled Student-t distributions by setting `family = scat`. Any improvement in AIC? How do the residuals look now?

5. Check whether a scaled Student-t with log-link function `scat(link=log)` achieves lower AIC. Then plot all the fitted effects of final model using `plot` (you can set `allTerms=TRUE` to plot also the parametric effects). Do the effects make sense?

2 Big GAM modelling of GEFCom14 electricity demand data

Here we use again data from the GEFCom14 challenge, but this data set is 24 times larger than the one used in the previous exercise. This is because it contains data corresponding to all the 24 hourly slots. The variable `Instant` indicates the hourly window corresponding to each row of the data set. All remaining covariates have the same interpretation as before.
Questions:

1. Load `mgcViz` and the data (`load("data/gefcom_big.rda")`). Create a model formula with smooth effects `wM`, `wM_s95`, `Instant`, `Trend` and `Posan`. Use regression splines bases (`bs='cr'`) for all smooths apart from `Posan`, for which you should use a cyclic basis (`bs='cc'`). Use `k = 6` for `Trend` and `k = 20` for `Posan`. Leave `k` to its default for the other smooths. Use parametric fixed effects for `Dow` and `NetDemand.24`. Use this formula within a `bamV` call to fit a Gaussian GAM. When calling `bamV` set `aGam=list(discrete=TRUE)` to speed up computations (do this in all subsequent `bamV` calls) and `aViz = list(nsim = 50)` to perform the response simulations needed for residuals checking. Having fitted the model, look at the effects using `plot` (recall that you can use argument `allTerms=TRUE` to plot also the parametric effects).

2. Use `check` to verify whether the number of basis functions used for the smooth effects is sufficiently large. Also, use the `check1D` function with the `l_gridCheck1D` layer to look for residual patterns across the variables.

3. Double `k` for any of the effects where the number of basis functions seems to small, and re-fit. After re-fitting, check whether AIC has improved and repeat the residual checks.

4. We expect that several of the effects might depend on the time of day. Use the `check2D` function with the `l_gridCheck2D` layer to look for interactions between `Instant` and `NetDemand.24`, `wM`, `wM_s95` and `Posan`. Notice that the binned mean residuals should ideally fall in the range (-2, 2) if the model was correct. Do you see any residual pattern? If so, fit a model which includes the necessary tensor product interactions (e.g. `ti(wM, Instant, k = c(10, 10))`) and repeat the checks. Are the patterns still there?

5. Assuming that we are now satisfied with our model, we'll now have a detailed look at the fitted smooth effects. First, look at the marginal effects using the `plot` function. Use the expression `print(plot(fit2, select = ???), pages = 1)` to plot all the marginal effects on one page (substitute `???` with the indexes of the univariate effects in your model). Do the same to plot the 2D interactions. Think about whether each effect makes physical sense to you. As an alternative to `plot`, recall that you can extract any effect using the `sm` function and produce a plot with customized layers. You can use the `listLayers` function to get a list of the available layers. Then, use the `plotRGL` function to manipulate each bivariate effect interactively.

6. **Extra question**: the model could be improved further. For instance, use the `check2D` function with the `l_gridCheck2D` layer to look at how the standard deviation and skewness of the residuals vary across pairs of covariates (the `e1071` package provides a `skewness` function, then you simply need to set `gridFun = skewness` in the call to `l_gridCheck2D`). Do you see any pattern? At this point we could consider a GAMLSS model with linear predictors for location, variance and skewness (e.g. the `gaulss` or `shash` family). However, `bam` methods does not yet support such models, so you'll need to use `gam` which can be much slower for large models.

3 Individual electricity demand modelling

Here we consider electriciy demand from 28 commercial costumers. The dataset covers roughly three months and it contains the following variables:

- `load` power usage from an individual costumer (in KW, I guess);
- `DateTime` the date and the time of day;
- `instant` the time of day, where 1 corresponds to 00:00-00:30, 2 to 00:30-01:00 and so on;
- `dow` factor variable indicating the day of the week;
- `temp` instantaneous temperature;
- `tempL` exponential smooth of `temp`, that is `tempL[i] = a*temp[i] + (1-a)*tempL[i-1]` with `a=0.95`;
- `ID` the unique ID of each individual costumer;
- `load48SM` lagged version of smoothed `load`, where the smoothing was performed as for `tempL`.

- `day` a counter depending on the day.

Questions:

1. Load `mgcViz` and the data (`load("data/Ind_elect.rda")`). Then use `bamV` to fit a Gaussian GAM model with smooth effects for `instant`, `temp` and `day`, and parametric effects for `dow` and `ID`. In the call to `bamV` set `aViz = list(nsim = 50)` to perform the response simulations needed for residuals checking. Look at the model output using `plot` and `summary`.

2. Now we start looking for interactions. Use the `check2D` function with the `l_gridCheck2D` layer to look for interactions between `ID` and `instant`, `temp` and `day`. Notice that the binned mean residuals should ideally fall in the range (-2, 2), if the model is correct. Do you see large deviation? If so for which costumer(s) in particular?

3. Modify the model formula to include by-factor smooths, that is `s(instant, by = ID, id = 1)` `s(temp, by = ID, id = 2)` and `s(day, by = ID, id = 3)`. The `id` argument make so that each of the 3 by-factor smooths has its own smoothing parameter, but the same smoothing parameter is used across all costumers. Refit the model using `bamV`, and set the argument `aGam=list(discrete=TRUE)` to speed up computation by discretisation. Compare this models to the previous one using AIC, and repeat the residuals checks. Any improvement?

4. Use `check` to verify whether the number of basis functions used for the smooth effects is sufficiently large. Double `k` for any of the effects where the number of basis functions seems to small, and re-fit. After re-fitting, check whether AIC has improved.

5. Use the `check2D` function with the `l_gridCheck2D` layer to look for interactions between `ID` and `load48SM`. If the effect of `load48SM` seems important, include the corresponding by-factor smooth by adding `s(load48SM, by = ID, id = 4)` to the model and re-fit.

6. Look at the model output using `plot`, using the `select` argument to plot any specific effect (you can't plot them all together, because the model includes tens of them). Compare the consumption of some of the individual costumers with the model predictions (which you can find in `fittedModel$fitted.values`). Do some costumers look much harder to predict than others?

4 Mackerel egg data

The following code loads and plots some data from a fish egg survey, for the purposes of spatial modelling.

```
library(mgcViz); load("data/mack.rda"); load("data/coast.rda")
## plot data....
with(mack,plot(lon,lat,cex=0.2+egg.dens/150,col="red"))
lines(coast)
ind <- c(1,3,4,5,10,11,16)
pairs(mack[,ind])
```

The main variables of interest in the `mack` data set are:

- `egg.count` number of eggs found in the net;
- `c.dist` distance from 200m seabed contour;
- `b.depth` depth of the ocean;

- `temp.surf` surface temperature of the ocean;
- `temp.20m` water temperature at a depth of 20 meters;
- `lat` latitude;
- `lon` longitude;
- `salinity`;
- `net.area` the area of the net used in m^2.

Questions:

1. Use the code above to load and plot the data;
2. Create a new variable `mack$log.net.area <- log(mack$net.area)`, and use `gamV` to fit a Poisson GAM with `egg.count` as response variable and 1D smooth effects for all the other variables, with the exceptions of `net.area` and `log.net.area`. Instead, include in the model formula the term `offset(log.net.area)`, meant to take into account the fact that the number of eggs captured is proportional to the net area.
3. Look at the model residuals using `qq`. What kind of problem do you see? Re-fit the models using a negative binomial (`family=nb`) or Tweedie (`family=tw`) response distribution, and check which model is better in terms of residuals QQ-plots and AIC.
4. Let `fit` be the best of the three GAM models you just fitted. Use `fit<-getViz(fit,nsim=50)` to get some simulated residuals, and then use the `check2D` function with the `l_gridCheck2D` layer to look for residual patters across `lon` and `lat`. Then refit the model using a bivariate isotropic effect `s(lon, lat, k=100)`, re-check the residuals and see whether AIC has improved.
5. Use `check` to verify whether the number of basis functions used for the smooth effects is sufficiently large. Then use the `check1D` function with the `l_gridCheck1D` layer look for residual patterns across some of the variables. If necessary, modify the model.
6. Plot the fitted effects using `plot`. Which effects look more important (look at the scales)? Use the `plotRGL` function to manipulate spatial effect interactively.

5 Bone mineral density modelling

This dataset is taken from the package `lava`. It consists of 112 girls randomized to receive calcium or placebo. The response variable of interests consists of longitudinal measurements of bone mineral density (g/cm^2) measured approximately every 6th month for 3 years. All girls are approximately 11yo at the start of the trial. The main variables are:

- `bmd` bone mass density;
- `group` placebo or supplement;
- `person` factor indicating the id of each girl;
- `age` the age of each girl at the time of each measurement;

Questions:

1. Load `mgcViz` and the data with `load("data/calcium.rda")`. Then use `gamV` to fit a Gaussian GAM model with `bmd` as response and linear effects for `age` and `group`. In the call to `gamV` set the argument `aViz=list(nsim = 50)` to have some simulated responses for residuals checks. Use `summary` to print the model output. Is the placebo effect significant? (which is the same as asking whether the treatment effect is significant)

2. Use `check1D` with the `l_gridCheck1D` layer to check that the mean of the negative residuals does not depart too much from 0, for any of the subjects. If you see significant departures add a random effect for `person` to the models formula (`s(person, bs="re")`), then re-fit and re-check the residuals. Print the model output again using `summary`.

3. Now modify the model formula to use a smooth effect for `age`, and plot the fitted effects using `plot`. Use the function `AIC` to compare the model with a smooth effects for `age` with the model which uses a linear `age` effect.

4. Verify whether the smooth age effect is different between the placebo and the treatment group, by using a by-factor smooth. To do this substitute `s(age)` with `s(age, by=group)` in the model formula, refit and then plot the fitted effects. To see the difference between the two smooths more clearly, use the `plotDiff` function with the `l_fitLine` and `l_ciLine` layers.

6 Retinopathy among diabetics

Data frame `wesdr` contains a subset of data from a Wisconsin study on development of retinopathy among diabetics. The following variables are provided:

- `ret`, a binary indicator of development of retinopathy by first follow up of study.
- `bmi`, the body mass index at entry to the study (between 18 and 25 is considered healthy).
- `dur`, the duration of diabetes, in years, at entry.
- `gly`, the percentage of glycosylated haemoglobin (HbA1C) in the blood (haemoglobin to which glucose has bound). 2.5-3.5% is normal for non-diabetics. 6.5% is generally considered good control for diabetics.

The aim is to model the probability of retinopathy as a function of the other variables.
Questions:

1. Load `mgcViz` and the data (`wesrd <- read.table("data/wesdr.txt")`), and use `pairs(wesdr)` to look at it.

2. It is not immediately clear what model structure for the linear predictor is appropriate. So in the first instance try all smooth main effects plus two way interactions using `ti` terms. Use a logistic regression model (`family = binomial`). In the call to `gamV` set `aViz = list(nsim = 50)` to do some residual simulations, and `aGam = list(select=TRUE)` to do some variable selection. Use `summary` and `plot` to verify which effects seems important.

3. Simplify the model by removing non-significant effects and re-fit. Check the model residuals using QQ-plots of normally transformed residuals (`qq(yourFit, type = "tnorm")`) as well as the `check1D` function with the `l_gridCheck1D` layer look for residual patterns across the individual variables.

4. Use the `plotRGL` function to visualize and manipulate any bi-variate effect in your model interactively.

7 CO_2 modelling

This question is about modelling data with seasonality, and the need to be very careful if trying to extrapolate with GAMs (or any statistical model). The data frame `co2s` contains monthly measurements of CO_2 at the south pole from January 1957 onwards. The columns are `co2`, the month of the year, `month`, and the cumulative number of months since January 1957, `c.month`. There are missing `co2` observations in some months.
Questions:

1. Load `mgcv` and the data with `library(gamair); data(co2s)`
2. Plot the CO_2 observations against cumulative months.
3. Fit a Gaussian additive model with a smooth effect for `c.month`, using the `gam` function. Use the `cr` basis, and a basis dimension of 100.
4. Obtain the predicted CO_2 for each month of the data, plus 36 months after the end of the data, as well as associated standard errors. Produce a plot of the predictions with twice standard error bands. Are the predictions in the last 36 months credible? NB: to produce the plot you have to write your own code, `mgcv` does not produce such plots.
5. Fit the model $\mathbb{E}(CO_2) = f_1(\texttt{c.month}_i) + f_2(\texttt{month}_i)$ where f_1 and f_2 are smooth functions. Use a basis of dimension 50 for f_1 and a cyclic basis for f_2. In the `gam` call, you will need to set argument `knots` to `list(month=c(1,13))` to make so that that the effect of January is the same as January, not that December and January are the same!
6. Repeat the prediction and plotting in question 4 for the new model. Are the predictions more credible now? Explain the differences between the new results and those from question 4.

8 Ozone modelling

Data frame `ozone` contains daily(ish) ozone measurements over Los Angeles (`O3`, ppm), along with:

`vh` the height at which the atmospheric pressure is 500mb, in metres.

`wind` the wind speed (reported as miles per hour, but this seems improbable).

`humidity` (usual % scale).

`temp` air temperature (Fahrenheit).

`ibh` the inversion layer base height in feet.

`ibt` the inversion base temperature (Fahrenheit).

`dpg` ‘Dagget air pressure gradient’ (mmhg).

`vis` visibility in miles.

`doy` Julian day, where 1 is Jan 1 1976.

The aim is to build a GAM model to explore the relationship between ozone and the other variables.
Questions:

1. Load the data using `ozone<-read.table("data/ozone.txt")` and use something like `pairs(ozone)` to look at it.

2. Load `mgcViz` and use `gamV` to fit a Gaussian GAM with `O3` as response, where $\log(\mathbb{E}(\texttt{O3}))$ is given by a sum of smooth functions (e.g. `s(wind)`) of each of the predictors. You will need to use the log-link, which requires using the argument `family=gaussian(link=log)` in the call to `gamV`. Plot the fitted effects using `plot`.

3. Check the model residuals using the `qq` and `check` functions. Do you see any residual pattern when you plot the residuals against the fitted values or linear predictor?

4. Refit the model using a Gamma as response distribution (`Gamma(link=log)`), and re-check the residuals. Does the residual distribution look better?

5. Fit an alternative model where you are using the identity-link (`Gamma(link=identity)`). Does a model with an additive (i.e. identity-link) structure do better than that with a multiplicative (log-link) structure in terms of `AIC`?

6. Plot the smoothed effects again and use the `summary` function to see which effects are significant. Try simplifying the model.

7. Once you have converged on a model, plot it and interpret the fitted smooth effects: do they make sense?

9 Larynx cancer in Germany

First load some data on cancer of the larynx by health reporting districts in Germany.

```
library(mgcViz)
load("data/german.polys.rda") ## load polygons defining German regions 'german.polys'
load("data/Larynx.rda")       ## load Larynx cancer death data 'Larynx'
## Get regions "midpoints"....
X <- t(sapply(german.polys,colMeans,na.rm=TRUE))
```

The variables in the `Larynx` dataframe are:

region code identifying region;

E expected number of deaths (according to population and pan German total);

Y number of deaths from Larynx cancer 1986-1990;

x measure of smoking rate in region.

Questions:

1. Run the code above and then use `gamV` to fit a Poisson GAM with a smooth effects for `x` and the following Markov Random Field (MRF) effect for region:

   ```
   s(region, k = 200, bs = "mrf", xt = list(polys=german.polys))
   ```

   and the offset term `offset(log(E))`, meant to take into account the fact that the number of death is proportional to the population of each region. In the call to `gamV` set `aViz = list(nsim = 50)` to do some residual simulations for later checking. Plot the fitted effects.

2. Now substitute the MRF smooth either with the isotropic smooth `s(X[,1],X[,2],k=200)` or with the tensor product smooth `te(X[,1],X[,2],k=c(15, 15))`. Plot the corresponding fitted effects and compare them. Which model does better in terms of AIC?

3. Use the `plotRGL` function to visualize and compare the fitted isotropic and tensor product smooth. You can use the code

   ```
   mfrow3d(1, 2)
   plotRGL(sm(fit2,1), residuals = TRUE)
   next3d()
   plotRGL(sm(fit3,1), residuals = TRUE)
   ```

   to get two rgl plots side by side.