R Workshop

Session 3:

Generalized Linear and Generalized Additive Models

Bill Venables, CSIRO, Australia

UseR! 2012

Nashville

11 June, 2012

Contents

1 An example from `MASS`: low birth weight

From Venables and Ripley (2002, Chap. 7).

low indicator of birth weight less than 2.5 kg.

age mother's age in years.

lwt mother's weight in pounds at last menstrual period.

race mother's race ('1' = white, '2' = black, '3' = other).

smoke smoking status during pregnancy.

ptl number of previous premature labours.

ht history of hypertension.

ui presence of uterine irritability.

ftv number of physician visits during the first trimester.

bwt birth weight in grams.

The original MASS code:

```
> attach(birthwt)
> race <- factor(race, labels = c("white", "black", "other"))
> table(ptl)
> ptd <- factor(ptl > 0)
> table(ftv)
> ftv <- factor(ftv)
> levels(ftv)[-(1:2)] <- "2+"
> table(ftv)  # as a check
> bwt <- data.frame(low = factor(low), age, lwt, race,
     smoke = (smoke > 0), ptd, ht = (ht > 0), ui = (ui > 0), ftv)
> detach(); rm(race, ptd, ftv)
```

My preference now:

```
> suppressPackageStartupMessages(library(SOAR))  # picky!
> suppressPackageStartupMessages(library(MASS))
> BirthWt <- within(birthwt, {
    race <- factor(race, labels = c("white", "black", "other"))
    ptl <- ptl > 0
    ftv <- factor(ftv)
    levels(ftv)[-(1:2)] <- "2+"
    low <- factor(low, labels = c("normal", "low"))
    smoke <- (smoke > 0)
    ht <- (ht > 0)
    ui <- (ui > 0)
    bwt <- NULL ## remove actual birth weight
  })
> Store(BirthWt)
> head(BirthWt, 2)

      low age lwt  race smoke   ptl    ht    ui ftv
85 normal  19 182 black FALSE FALSE FALSE  TRUE   0
86 normal  33 155 other FALSE FALSE FALSE FALSE  2+
```

Advice from the `fortunes` package:

> *If I were to be treated by a cure created by stepwise regression, I would prefer voodoo.*
>
> — Dieter Menne *(in a thread about regressions with many variables) R-help (October 2009)*

- Automated screening is more defensible in cases of pure prediction.
- Automated screening is dangerous if used for inference.

Caveat emptor!

We reduce some of the clutter with:

```
> options(show.signif.stars = FALSE)
> stepAIC <- function(..., trace = FALSE)  ## change default
      MASS::stepAIC(..., trace = trace)
> dropterm <- function(..., sorted = TRUE) ## change default
      MASS::dropterm(..., sorted = sorted)
```

1.1 Automated screening of variables

A starting point, main effects only:

```
> BW0 <- glm(low ~ ., binomial, BirthWt)
> dropterm(BW0, test = "Chisq")

Single term deletions
Model:
low ~ age + lwt + race + smoke + ptl + ht + ui + ftv
       Df Deviance    AIC    LRT   Pr(Chi)
ftv     2   196.83 214.83 1.3582 0.507077
age     1   196.42 216.42 0.9419 0.331796
<none>      195.48 217.48
ui      1   197.59 217.59 2.1100 0.146342
smoke   1   198.67 218.67 3.1982 0.073717
race    2   201.23 219.23 5.7513 0.056380
lwt     1   200.95 220.95 5.4739 0.019302
ht      1   202.93 222.93 7.4584 0.006314
ptl     1   203.58 223.58 8.1085 0.004406
```

Screen for possible interactions:

```
> sBW0 <- stepAIC(BW0, scope = list(lower = ~1, upper = ~.^2))
> dropterm(sBW0, test = "Chisq")

Single term deletions
Model:
low ~ age + lwt + smoke + ptl + ht + ui + ftv + age:ftv + smoke:ui
         Df Deviance    AIC     LRT   Pr(Chi)
<none>        183.07 207.07
smoke:ui  1   186.99 208.99  3.9127 0.0479224
ht        1   191.21 213.21  8.1374 0.0043361
lwt       1   191.56 213.56  8.4856 0.0035797
ptl       1   193.59 215.59 10.5146 0.0011843
age:ftv   2   199.00 219.00 15.9295 0.0003475
```

1.2 An extended model with smooth terms

We consider some flexibility in the *age* term and its interaction with *ftv*.

```
> suppressPackageStartupMessages(require(mgcv))
> BW1 <- gam(low ~ smoke*ui + ht + s(lwt) + ptl + s(age) +
             poly(age, 2)*ftv, family = binomial, data = BirthWt)
```

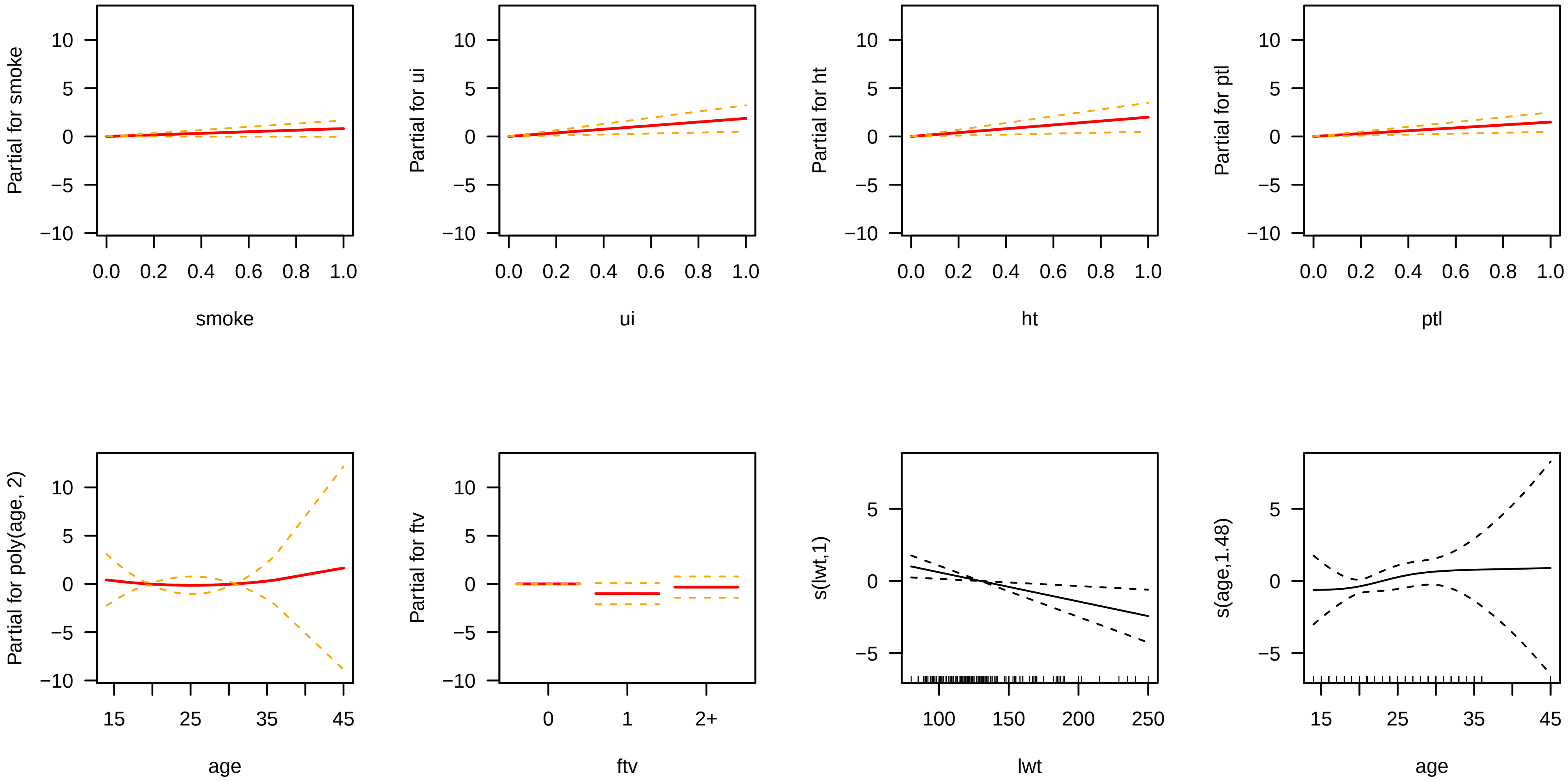
```
> anova(BW1)
Family: binomial
Link function: logit
Formula:
low ~ smoke * ui + ht + s(lwt) + ptl + s(age) + poly(age, 2) *
    ftv
Parametric Terms:
                 df Chi.sq p-value
smoke             1  3.764 0.05236
ui                1  7.563 0.00596
ht                1  7.095 0.00773
ptl               1  8.846 0.00294
poly(age, 2)      1  0.252 0.61580
ftv               2  3.434 0.17961
smoke:ui          1  3.860 0.04945
poly(age, 2):ftv  4 13.389 0.00952
Approximate significance of smooth terms:
         edf Ref.df Chi.sq p-value
s(lwt) 1.000  1.000  7.044 0.00796
s(age) 1.479  1.866  1.262 0.49144
```

1.3 Looking at the terms

First the "main effect" terms. This is a bit tricky...

```
> (nam <- names(model.frame(BW1)))
[1] "low"          "smoke"        "ui"           "ht"
[5] "ptl"          "poly(age, 2)" "ftv"          "lwt"
[9] "age"
> layout(matrix(1:8, 2, 4, byrow = TRUE))
> termplot(BW1, terms = nam[2:7], se=TRUE) ## fixed
> plot(BW1)                                ## smooth
```

The first *nam* is the response and the 8th and 9th refer to the smooth terms. *termplot* can handle non-smoothed terms, but smooth terms must be handled by the *plot* method for *gam* objects.


Partial for smoke
smoke
Partial for ui
ui
Partial for ht
ht
Partial for ptl
ptl
Partial for poly(age, 2)
age
Partial for ftv
ftv
0
1
2+
s(lwt,1)
lwt
s(age,1.48)
age

1.4 A helper function: the most frequent value

```
> mostFreq <- function(x, ...) UseMethod("mostFreq")
> mostFreq.numeric <- stats::median.default  ## check argument names
> mostFreq.logical <- function(x, ...) {
    tx <- as.vector(table(x))
    tx[2] > tx[1]
  }
> mostFreq.character <- function(x, ...) {
    tx <- table(x)
    names(tx)[which.max(tx)]
  }
> mostFreq.factor <- function(x, ...)
      mostFreq.character(as.character(x))
> Store(list = ls(pattern = "^mostFreq"))
```

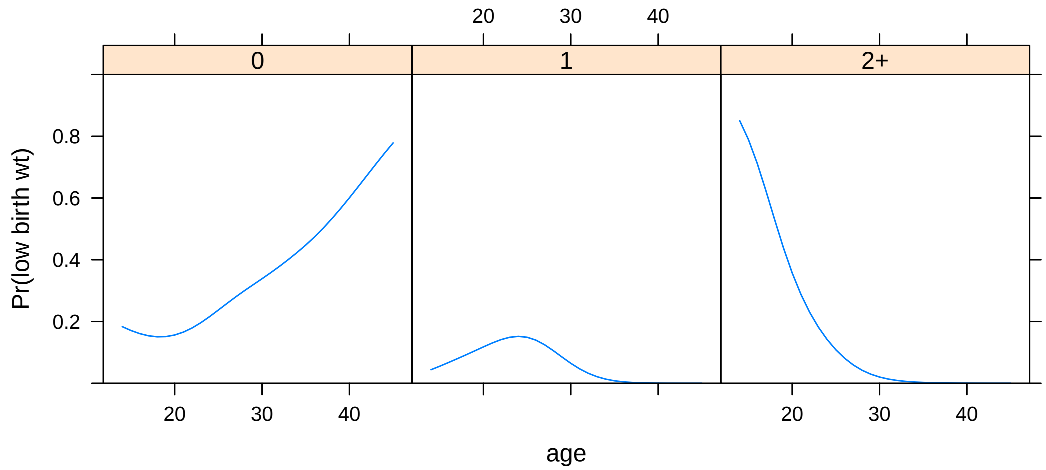
1.5 The main two-way interaction

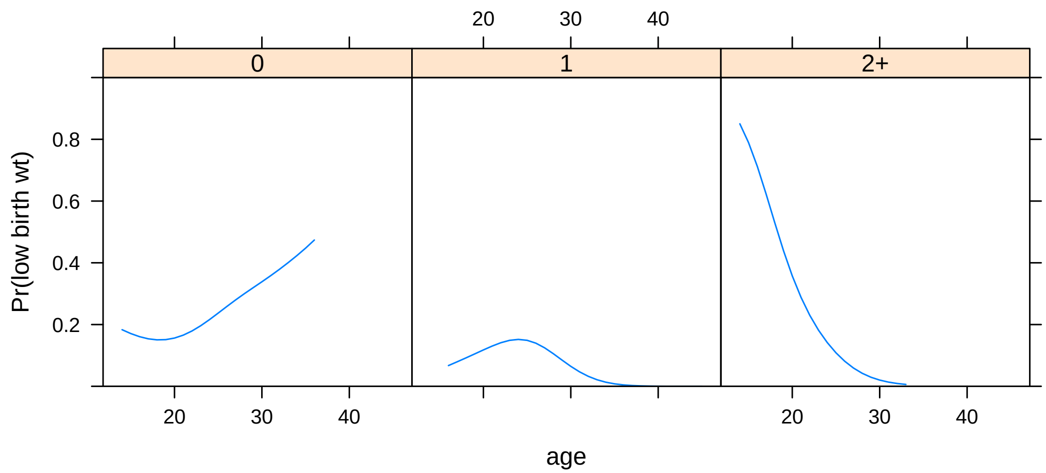
Predict the probability of low birth weight with varying *age* and *ftv*, and other variables at or near their modal value.

```
> all.vars(formula(BW1))
[1] "low"   "smoke" "ui"    "ht"    "lwt"   "ptl"   "age"   "ftv"
> pBirthWt <- with(BirthWt,
      expand.grid(smoke = mostFreq(smoke), ui = mostFreq(ui),
                  ht = mostFreq(ht), lwt = mostFreq(lwt),
                  ptl = mostFreq(ptl), age = min(age):max(age),
                  ftv = levels(ftv)))
> pBirthWt$pBW1 <- predict(BW1, pBirthWt, type = "response")
> library(lattice)
> (ageFtv <- xyplot(pBW1 ~ age|ftv, pBirthWt, layout = c(3,1),
                    type = "l", ylab = "Pr(low birth wt)",
                    ylim = 0:1, aspect = 1))
```

To bring the predictions closer to the actual *data*, confine the predictions to `age` ranges that apply within the levels of `ftv`

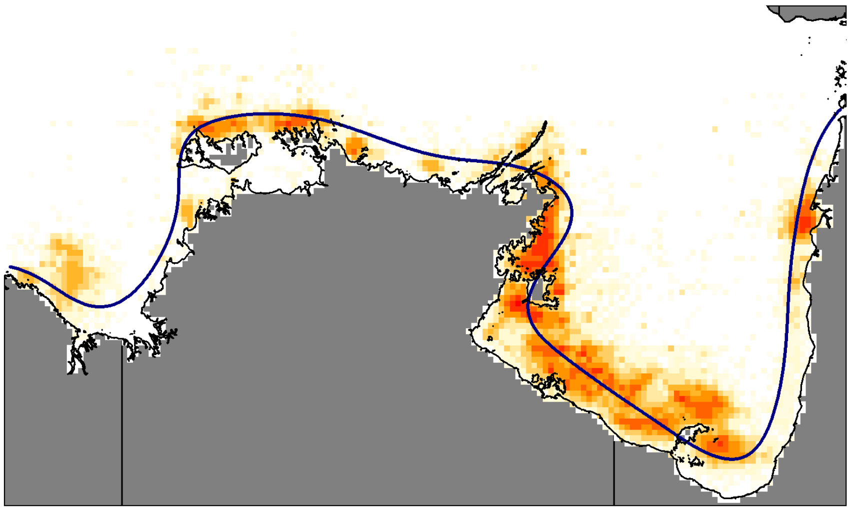
```
> pBirthWt <- within(pBirthWt, {
      rngs <- do.call(cbind, with(BirthWt, tapply(age, ftv, range)))
      pBW1a <- pBW1
      is.na(pBW1a[age < rngs[1, ftv] | age > rngs[2, ftv]]) <- TRUE
      rm(rngs)
  })
> (ageFtvA <- xyplot(pBW1a ~ age|ftv, pBirthWt, layout = c(3,1),
                     type = "l", ylab = "Pr(low birth wt)",
                     ylim = 0:1, aspect = 1))
```


0
1
2+
Pr(low birth wt)
0.2
0.4
0.6
0.8
20
30
40
age


0
1
2+
Pr(low birth wt)
0.2
0.4
0.6
0.8
20
30
40
age

2 Tiger prawn species split

The Northern Prawn Fishery: Tiger prawn effort and “the blue line”



Background

- Two species of Tiger prawns are caught together.
- Both species require separate Stock Assessment.
- The assessment model requires catches of Tiger prawns to be split (by weight)
- Problem: Build a model for partitioning catches into the two compoent species.
- Data: independent surveys (12 in all) where catches have been split into the two species, *Penaeus semisulcatus* (Grooved) and *P. esculentus* (Brown).
- Both species have annual offshore migration patterns.

Variables available:

- Response: $\mathtt{Psem}$, $\mathtt{Pesc}$, $(\mathtt{Total} = \mathtt{Psem} + \mathtt{Pesc})$ in gms;

  Predictors:

- $\mathtt{Longitude}$, $\mathtt{Latitude}$ – of trawl shot;
- $\mathtt{Coast}$, $\mathtt{Sea}$ – alternative spatial coordinates;
- $\mathtt{Depth}$ – of trawl shot;
- $\mathtt{Mud}$ – the % mud in the substrate;
- $\mathtt{DayOfYear}$ – to allow for annual migration periodicity;
- $\mathtt{ElapsedDays}$ – days since 1970-01-01, for long term trend
- $\mathtt{Survey}$ – used for a random effect extension.

Strategy:

- Build a simpler GLM using mainly splines, with a term in `DayOfYear` and `Sea` to allow for temporal (annual migration) effects.
- Develop a more sophisticated GAM to take advantage of more recent modelling technology
- Look at a long-term trend term as a perturbation to the model
- Consider GLMMs with random terms for `Survey`, eventually

Model terms, GLM:

- Spline in `Coast` surrogate for large-scale benthic changes,
- Splines in `Sea`, `Depth` and `Mud` – more local spatial effects,
- Periodic term in `DayOfYear` and its interaction with `Sea` – annual migration effects,
- Spline in `ElapsedDays` – testing for long-term stability.

2.1 An initial GLM

The GLM fitting process is slow to converge under the normal algorithm. Two possible alternatives:

- Use the `glm2` library, which has a modified convergence process, (Marschner, 2011).
- In this case the problem is with the variable weights needed. Fit a model ignoring weights, and use the linear predictor as a starting value for the weighted fit.

The periodic terms will use Fourier polynomials:

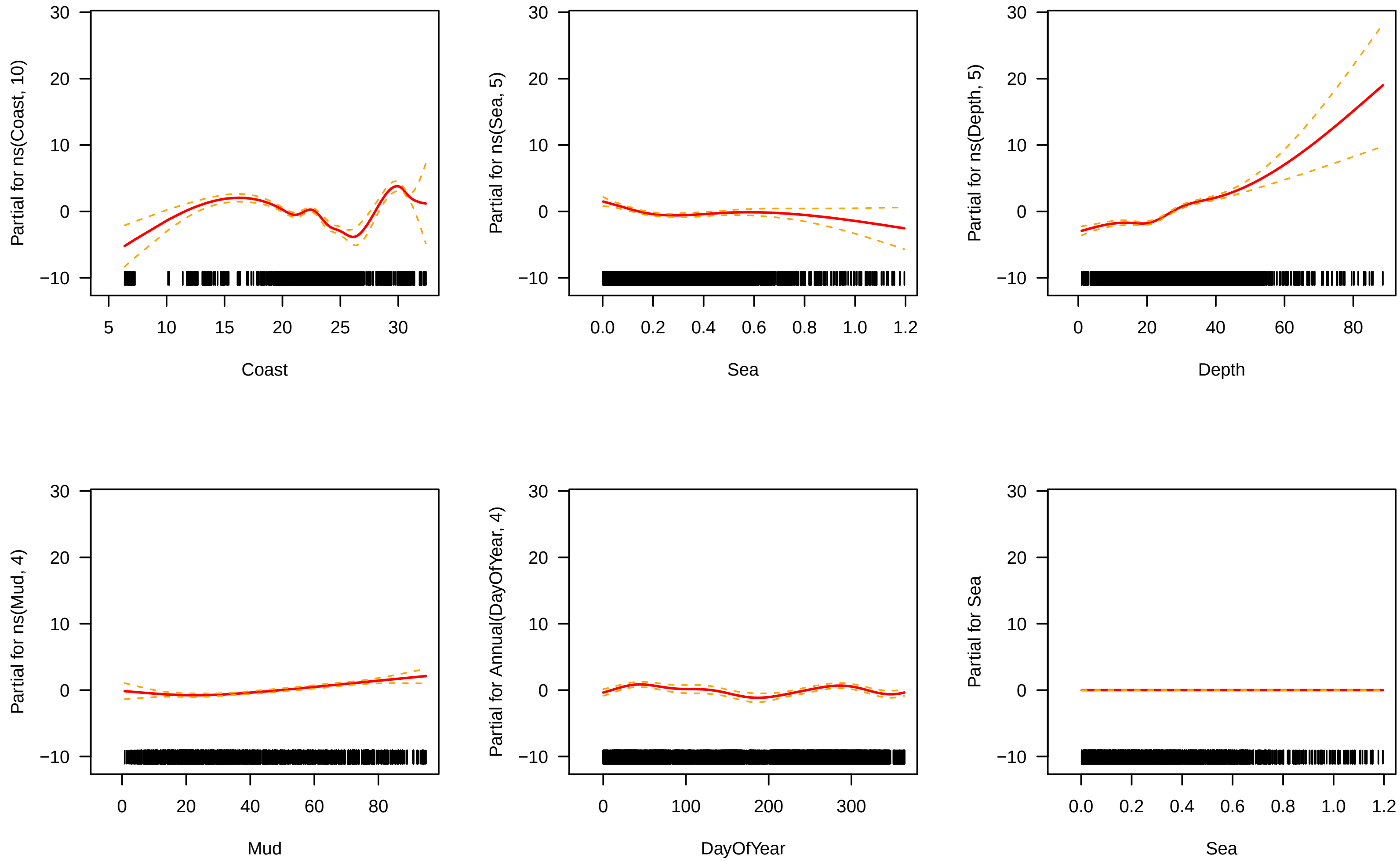
```
> Annual

function (day, k = 4) {  ## day of the year, starting from 0
    theta <- 2*base::pi*day/364.25
    X <- matrix(0, length(theta), 2 * k)
    nam <- as.vector(outer(c("c", "s"), 1:k, paste, sep = ""))
    dimnames(X) <- list(names(day), nam)
    m <- 0
    for (j in 1:k) {
        X[, (m <- m + 1)] <- cos(j * theta)
        X[, (m <- m + 1)] <- sin(j * theta)
    }
    X
}
```

```
> library(splines)
> temp <- glm(Psem/Total ~ ns(Coast, 10) + ns(Sea, 5) + ns(Depth, 5) +
              ns(Mud, 4) + Annual(DayOfYear, 4)*Sea, family = quasibinomial,
              data = Tigers, trace = TRUE)  ## unweighted
Deviance = 4911.782 Iterations - 1
Deviance = 4361.232 Iterations - 2
Deviance = 4292.588 Iterations - 3
Deviance = 4290.277 Iterations - 4
Deviance = 4290.268 Iterations - 5
Deviance = 4290.268 Iterations - 6
> Tigers$eta <- predict(temp)
```

```
> TModelGLM <- update(temp, etastart = eta, weights = Total)
Deviance = 5487650 Iterations - 1
Deviance = 5435326 Iterations - 2
Deviance = 5434241 Iterations - 3
Deviance = 5434238 Iterations - 4
Deviance = 5434238 Iterations - 5
> rm(temp)
> Tigers$eta <- predict(TModelGLM)
> TModelGLM$call$trace <- NULL  ## for future updating
> Store(TModelGLM)
> (nam <- names(model.frame(TModelGLM)))  ## for term plotting
[1] "Psem/Total"            "ns(Coast, 10)"        "ns(Sea, 5)"
[4] "ns(Depth, 5)"          "ns(Mud, 4)"           "Annual(DayOfYear, 4)"
[7] "Sea"                   "(weights)"            "(etastart)"
> nam <- nam[2:7]                        ## terms to plot
```

Look at the shape of the main effect terms, to see implications:

```
> layout(matrix(1:6, 2, 3, byrow=TRUE))   ## 2 x 3 array of plots
> termplot(TModelGLM, terms = nam, se = TRUE, rug=TRUE)
```


Partial for ns(Coast, 10)
Coast
Partial for ns(Sea, 5)
Sea
Partial for ns(Depth, 5)
Depth
Partial for ns(Mud, 4)
Mud
Partial for Annual(DayOfYear, 4)
DayOfYear
Partial for Sea
Sea

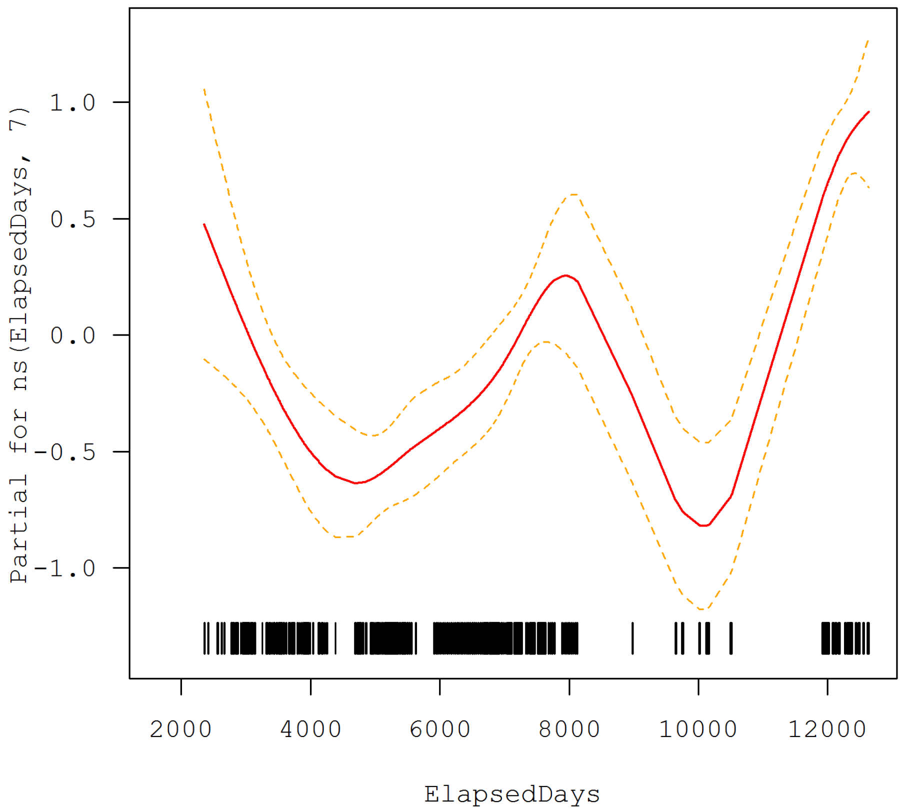
2.2 A long-term trend?

The stability of species ratios over time is important. We can check for this by including a spline term in *ElapsedDays*:

```
> TM2 <- update(TModelGLM, . ~ . + ns(ElapsedDays, 7))
> anova(TModelGLM, TM2, test = "F")

Analysis of Deviance Table
Model 1: Psem/Total ~ ns(Coast, 10) + ns(Sea, 5) + ns(Depth, 5) + ns(Mud,
    4) + Annual(DayOfYear, 4) * Sea
Model 2: Psem/Total ~ ns(Coast, 10) + ns(Sea, 5) + ns(Depth, 5) + ns(Mud,
    4) + Annual(DayOfYear, 4) + Sea + ns(ElapsedDays, 7) + Annual(DayOfYear,
    4):Sea
  Resid. Df Resid. Dev Df Deviance      F    Pr(>F)
1     12528    5434238
2     12521    5005830  7   428408 25.618 < 2.2e-16
```

Significant, but is it important?

```
> termplot(TM2, terms = "ns(ElapsedDays, 7)", se=TRUE, rug = TRUE)
```



2.3 A working GAM with new technology

The `mgcv` package represents a major advance in smooth model fitting technology in sevaral respects, including

- Smoothed terms in multiple predictors can now be handled,
- A wide variety of basis functions is available, including e.g. thin plate splines, cyclic spline bases, &c,
- A powerful *visualisation* tool in projections of predictor variable space is avialable in addition to tools for inspection of individual terms

The price is:

- The package is still under development and new versions are fairly common (though becoming less so)
- The implementation is to some extent non-standard **R**.

The working model:

```
> require(mgcv)
> Attach()
> TModelGAM <- gam(Psem/Total ~ s(Longitude, Latitude) +
                   te(DayOfYear, Sea, k = c(5, 5), bs = c("cc", "cs")) +
                   te(DayOfYear, Depth, k = c(5, 5), bs = c("cc", "cs")) +
                   te(Sea, Depth, k = c(5, 5), bs = "cs") +
                   s(Mud, k = 5),
                   family = quasibinomial, data = Tigers,
                   knots = list(DayOfYear = seq(0, 365.25, length=5)),
                   weights = Total, control = gam.control(trace = TRUE))
> TModelGAM_NS <- update(TModelGAM, . ~ . + s(ElapsedDays))  ## non-stationary
```

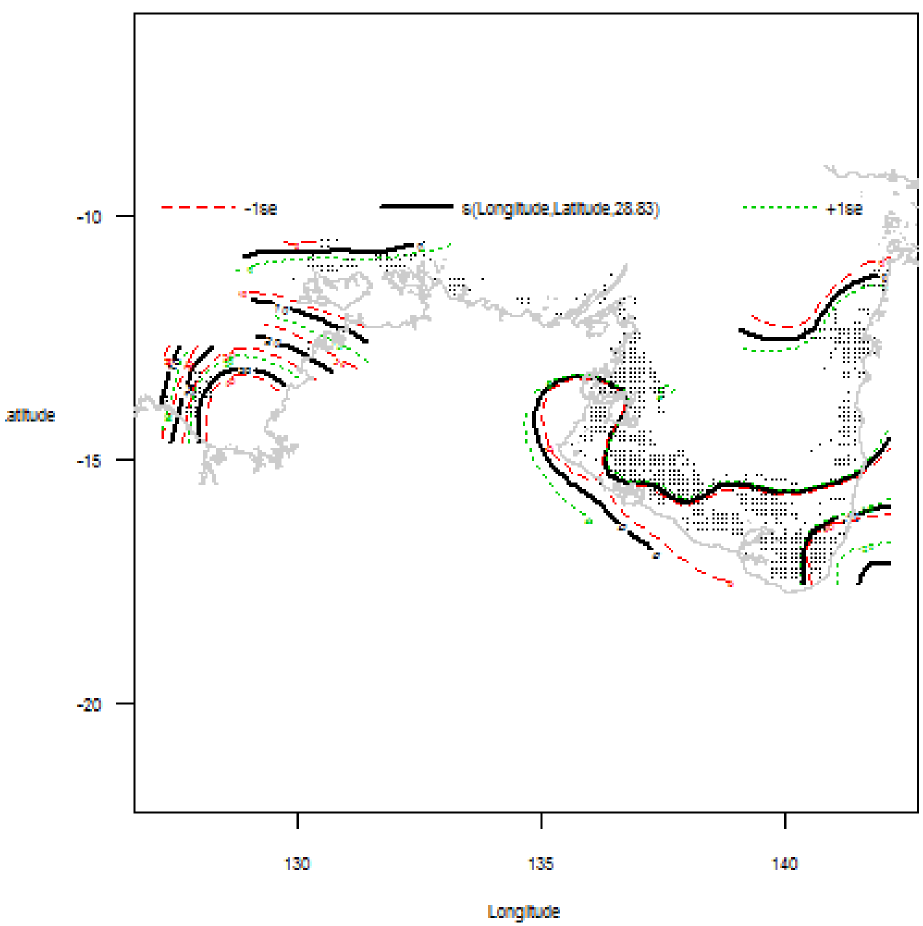
Some notes:

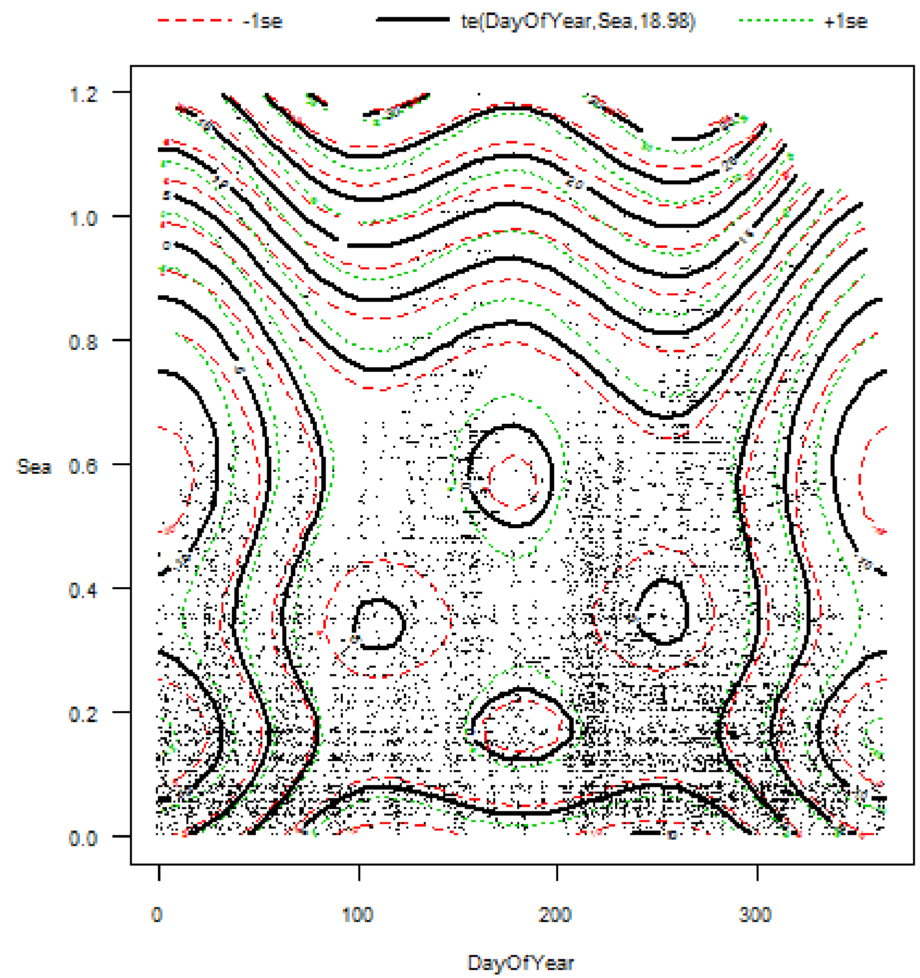
- An *isotropic* spatial term in `Longitude` and `Latitude` to account for purely spatial effects;
- Tensor spline (non-isotropic) terms in each pair of `DayOfYear`, `Depth` and `Sea` to account for temporal and environmental effects;
- A smooth term in `Mud`, also for environmental effects;
- The terms in `DayOfYear` are periodic with period 365 days (guaranteed by the data) in that coordinate `bs = "cc"`;
- Other terms use a smooth spline basis, `bs = "cs"`. Other choices of bases are available.

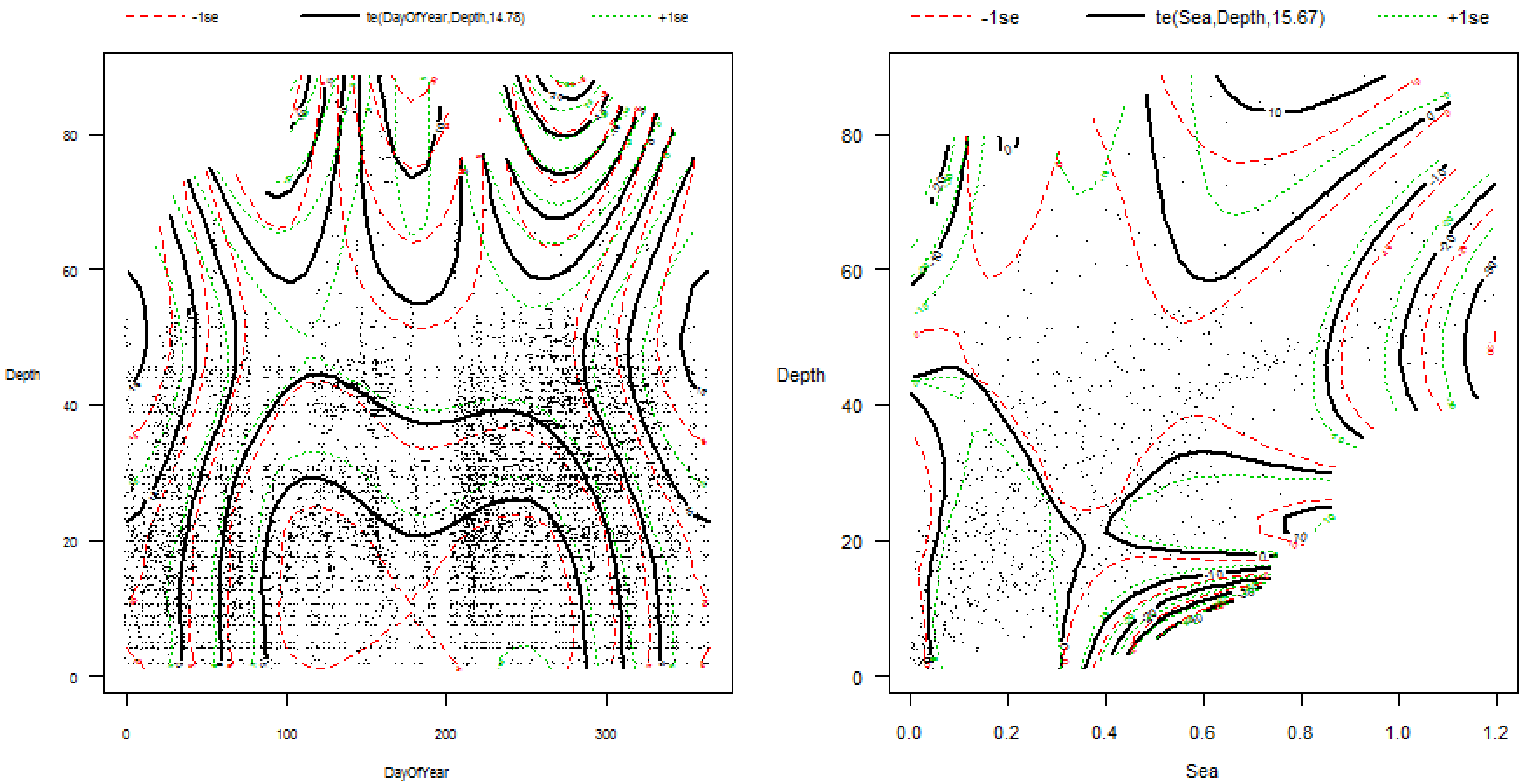
Some views of the fit:

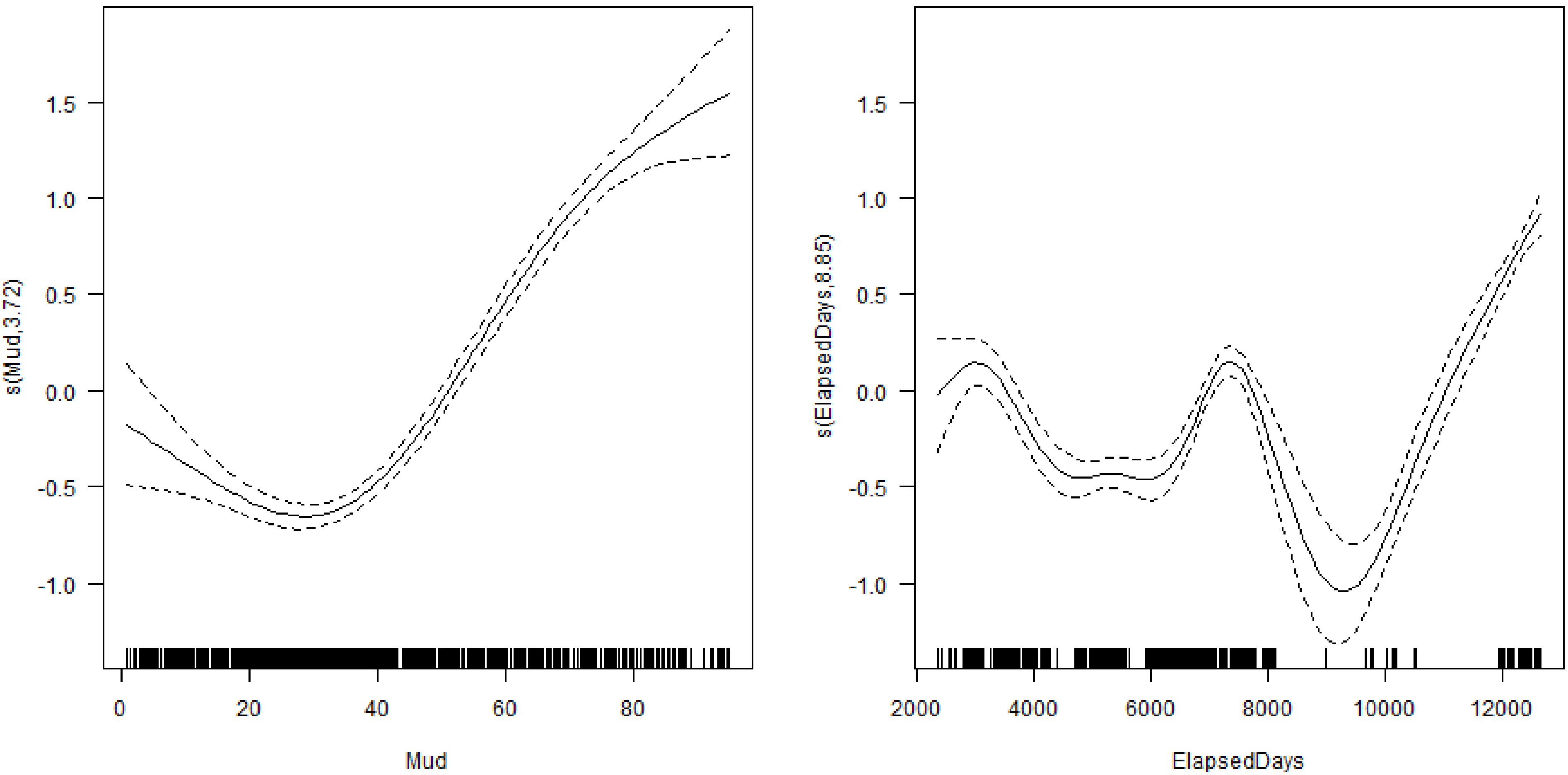
```
> png(file = "Fig/03tm2_%03d.png", height = 500, width = 900)
> par(las = 1)
> layout(rbind(1:2))
> plot(TModelGAM_NS, select = 1, asp = 1)
> lines(Oz, col = grey(0.8))
> title(main = "Lon x Lat, isotropic")
> for(j in 2:6)
      plot(TModelGAM_NS, select = j)
> vis.gam(TModelGAM_NS, view = c("Longitude","Latitude"))
> title(main = "Lon x Lat, isotropic")
> vis.gam(TModelGAM_NS, view = c("DayOfYear","Sea"))
> title(main = "Day of year x Sea")
> vis.gam(TModelGAM_NS, view = c("DayOfYear","Depth"))
> title(main = "Day of year x Depth")
> vis.gam(TModelGAM_NS, view = c("Depth","Mud"))
> title(main = "Depth x Mud")
> dev.off()
```

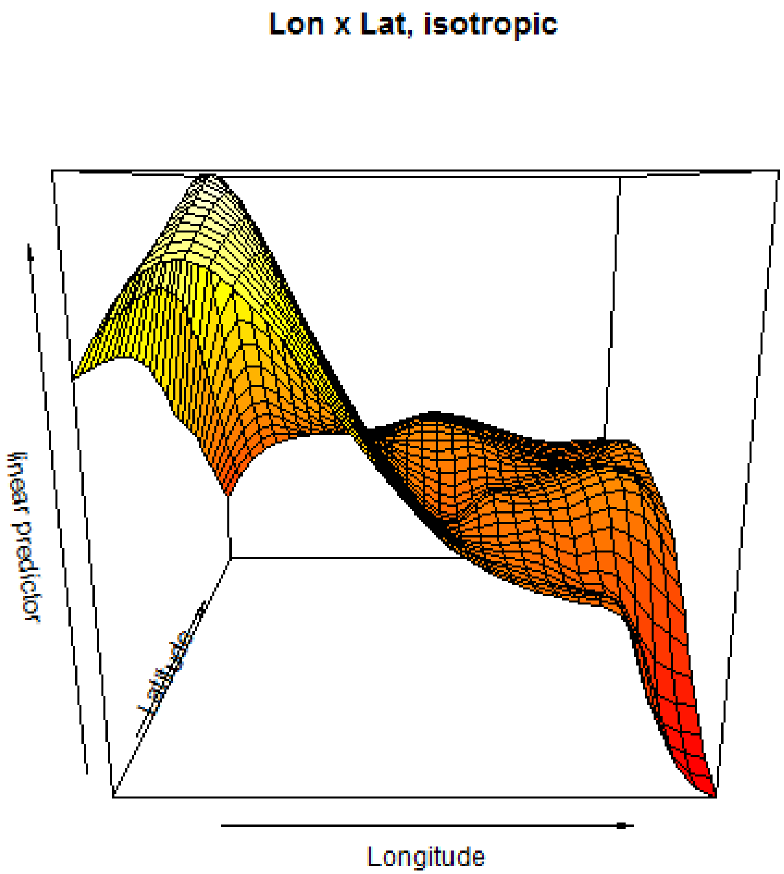
Lon x Lat, isotropic

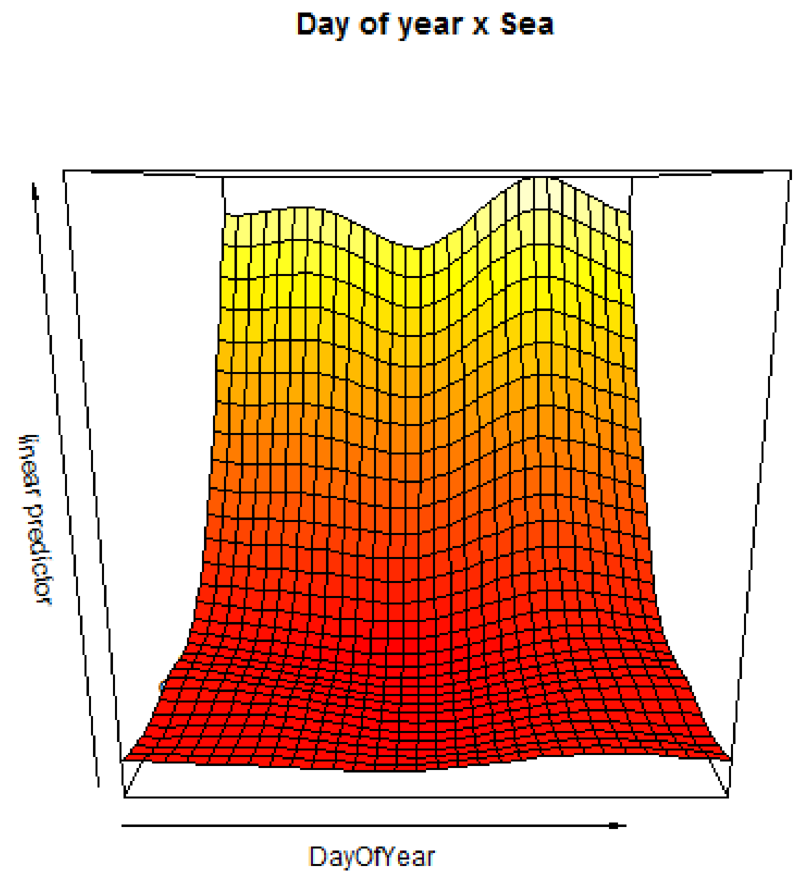


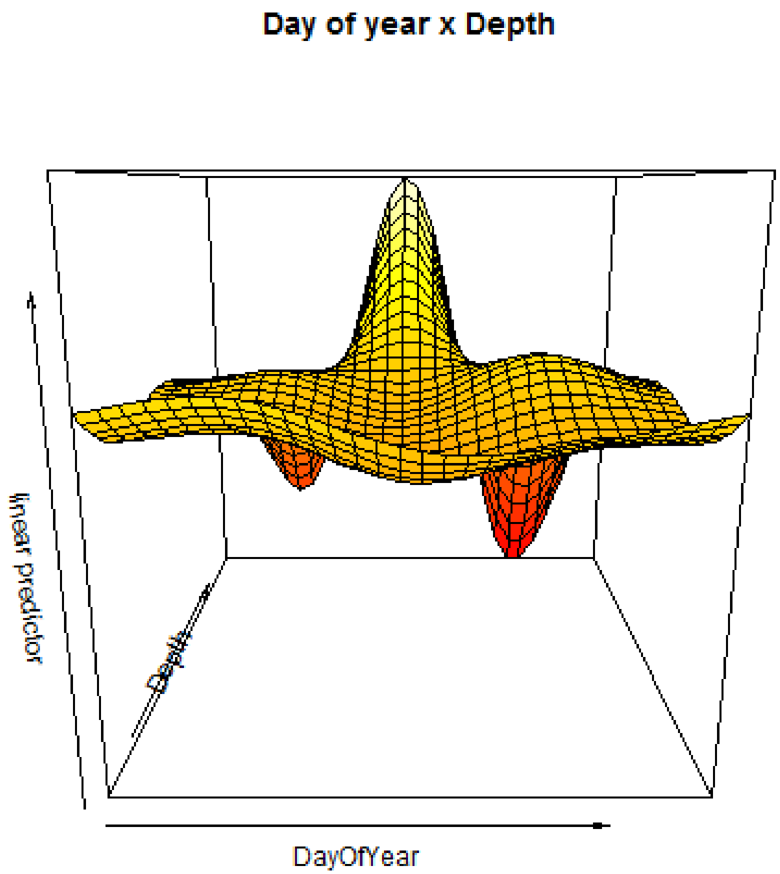



-1se
te(DayOfYear,Depth,14.78)
+1se
Depth
DayOfYear
0
100
200
300
0
20
40
60
80
-1se
te(Sea,Depth,15.67)
+1se
Depth
Sea
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
20
40
60
80

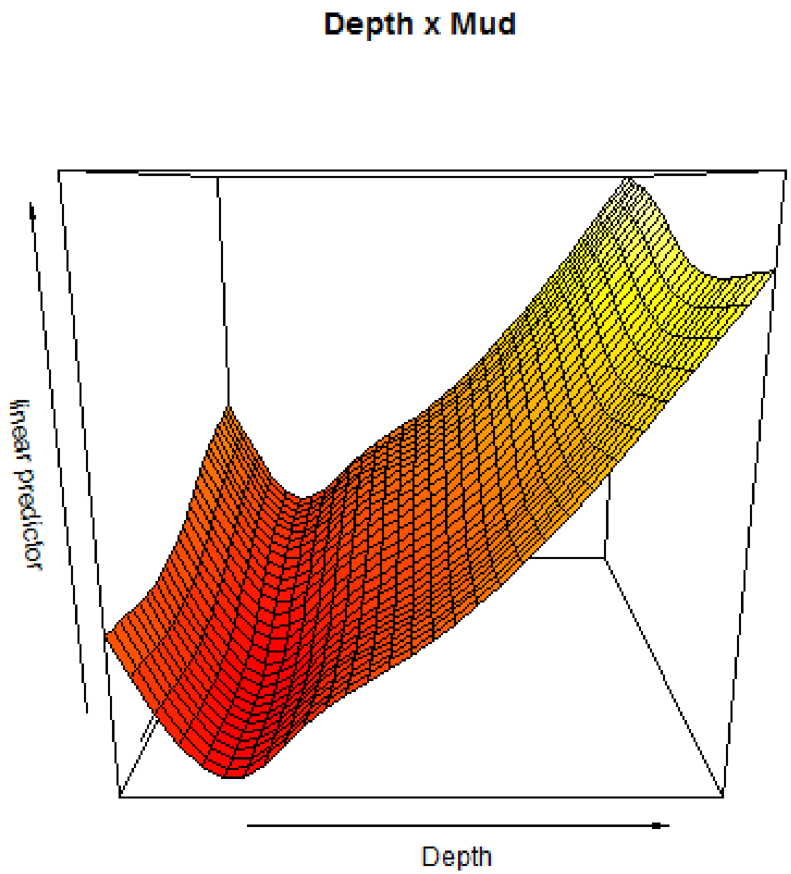

s(Mud,3.72)
1.5
1.0
0.5
0.0
-0.5
-1.0
0
20
40
60
80
Mud
s(ElapsedDays,8.85)
1.5
1.0
0.5
0.0
-0.5
-1.0
2000
4000
6000
8000
10000
12000
ElapsedDays


Lon x Lat, isotropic
linear predictor
Latitude
Longitude


Day of year x Sea
linear predictor
DayOfYear



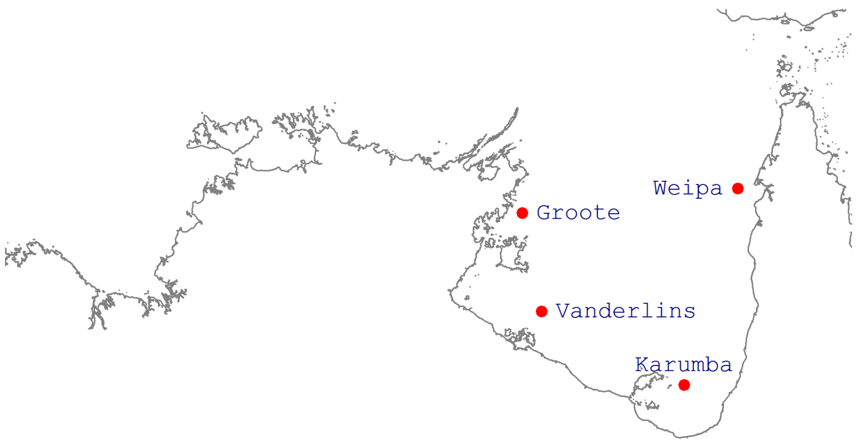
Day of year x Depth
linear predictor
Depth
DayOfYear

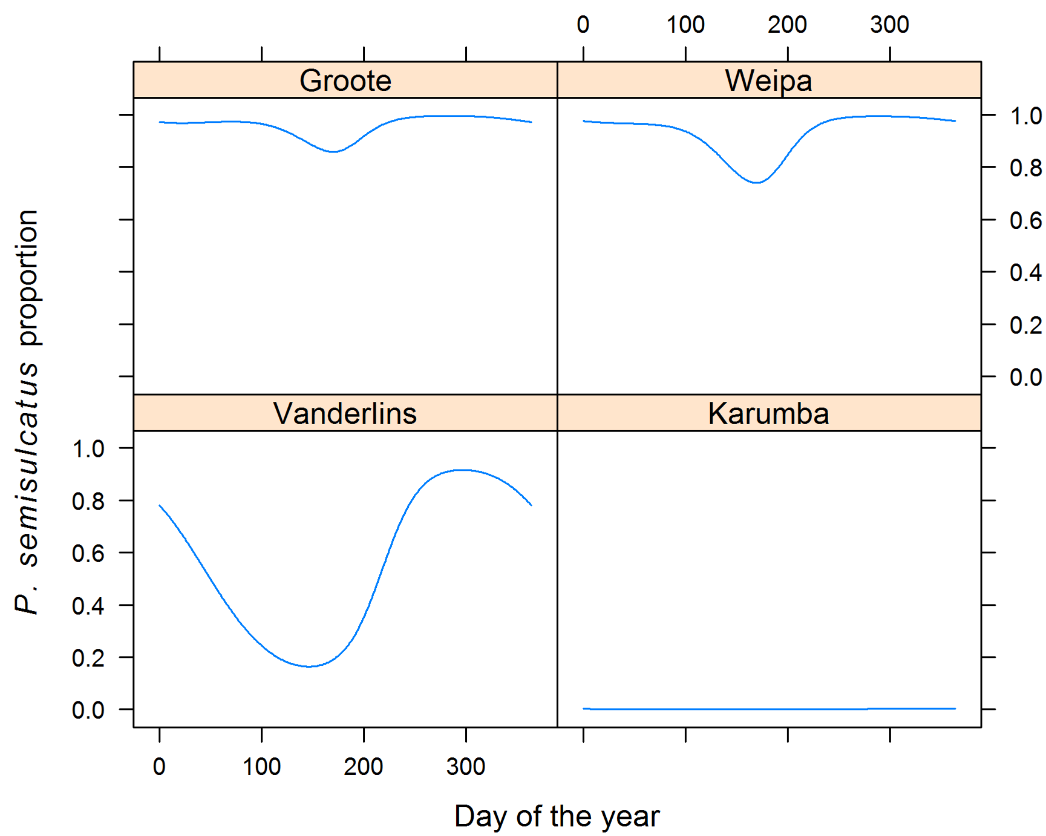


Depth x Mud
linear predictor
Depth

2.4 The spatio-temporal effect

Finally, we look at daily predictions for one year at 4 locations within the Gulf of Carpentaria:





Groote
Weipa
Vanderlins
Karumba
0
100
200
300
1.0
0.8
0.6
0.4
0.2
0.0
P. semisulcatus proportion
Day of the year

Note that the migration effect is strongest in the Vanderlins islands region, where the Tiger prawn catch is high.

The prediction code is not shown, but the results are stored in a data frame called *Data4*. The graphic is generaged by:

```
> Attach()
> require(lattice)
> print(xyplot(Fsemi ~ DayOfYear|Place, Data4, type = "l",
    ylab=expression(italic(P.)~~italic(semisulcatus)~~plain(proportion)),
    xlab = "Day of the year", aspect = 0.7))
```

Note the device for mixed fonts in the annotations.

3 Technical highlights

- Slide 5: Using *`within`* for neat data manipulation.
- Slide 6: Creating a temporary override function with changed defaults.
- Slide 8: *`stepAIC`* and *`dropterm`*.
- Slide 9: Suppressing package startup messages; simple *`gam`*.
- Slide 11: *`termplot`* and finding the terms that can be plotted.
- Slide 13: Elementary `S3` generic and methods.
- Slide 14: *`all.vars`* finding all variables in an expression.
- Slide 22: The `glm2` package for stable GLM fitting.
- Slide 24: Coddling a troublesome GLM fit with starting values.
- Slide 31 *et seq.*: Sophisticated GAM modelling and visualisation.

References

Marschner, I. C. (2011, December). glm2: Fitting generalized linear models with convergence problems. *The* **R** *Journal 3*(2), 12–15.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with* **S** (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.

Session information

- R version 2.15.0 (2012-03-30), `i386-pc-mingw32`
- Locale: `LC_COLLATE=English_Australia.1252`, `LC_CTYPE=English_Australia.1252`, `LC_MONETARY=English_Australia.1252`, `LC_NUMERIC=C`, `LC_TIME=English_Australia.1252`
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, utils
- Other packages: lattice 0.20-6, MASS 7.3-18, mgcv 1.7-17, PBSmapping 2.62.34, SOAR 0.99-10
- Loaded via a namespace (and not attached): grid 2.15.0, Matrix 1.0-6, nlme 3.1-104, tools 2.15.0