Harvard Department of Government 2003
Faraway Chapter 15, Additive Models

JEFF GILL
Visiting Professor, Fall 2024

Generalized Additive Models

- Big Picture: just like a GLM except we will do component-wise smoothing of some right-hand side variables.
- More computationally intensive that GLM estimation with many more model-fitting choices to make.
- Results are often given graphically for smoothed parameters, especially if there are many.
- Definitive citations:
  - Hastie and Tibshirani (1986), "Generalized Additive Models" (with discussion). *Statistical Science* 1, 297-318.
  - Wood (2006), *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC.
  - Hastie (1993), in Chambers and Hastie, *Statistical Models in S*. Chapman & Hall.
  - Hastie and Tibshirani (1990), Generalized Additive Models. Chapman & Hall.

Generalized Additive Models

- Structure:

$$Y = \alpha + \sum_{j=1}^{n} f_j(x_j) + \epsilon$$

$$E[\epsilon] = 0$$

$$\mathrm{cor}(\epsilon_i, x_j) = 0$$

$$\mathrm{Var}(\epsilon) = \sigma^2$$

- Solved by an algorithm called "backfitting."
- Typically we think of f_j's as univariate and smooth, but they don't have to be either: $f(x_{j1}, x_{j2})$ like an interaction or other single dimension mapping, or categorical specifications.

Generalized Additive Models

- To avoid a plethora of free constants in each of the $f_j()$, it is common to assume $E[f_j(x_j)] = 0$, which can be achieved by centering if necessary.
- *Big point*: unlike a GLM, each term is represented additively and therefore we can use the same marginal interpretation as linear models (but without the linear assumption obviously). Two consequences:
  1. The variation of the fitted response surface holding all but one explanatory variable constant does not depend on the values of the other explanatory values.
  2. Plots of the fits separately are very useful.

Botanical Example

- Here is a *standard* example concerning cherry trees, which are less linear than one would think.
- The simple model of interest is:

$$\log(\texttt{Volume}_i) = f_1(\texttt{Height}_i) + f_2(\texttt{Girth}_i) + \epsilon_i.$$

- Start with:

```
library(mgcv)
data(trees)
t(trees)
```

```
       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18]
Girth   8.3  8.6  8.8 10.5 10.7 10.8 11.0 11.0 11.1  11.2  11.3  11.4  11.4  11.7  12.0  12.9  12.9  13.3
Height 70.0 65.0 63.0 72.0 81.0 83.0 66.0 75.0 80.0  75.0  79.0  76.0  76.0  69.0  75.0  74.0  85.0  86.0
Volume 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6  19.9  24.2  21.0  21.4  21.3  19.1  22.2  33.8  27.4

       [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31]
Girth   13.7  13.8  14.0  14.2  14.5  16.0  16.3  17.3  17.5  17.9  18.0    18  20.6
Height  71.0  64.0  78.0  80.0  74.0  72.0  77.0  81.0  82.0  80.0  80.0    80  87.0
Volume  25.7  24.9  34.5  31.7  36.3  38.3  42.6  55.4  55.7  58.3  51.5    51  77.0
```

Botanical Example

- Volume must be positive, so apply a Gamma link function:

```
tree.gam.1 <- gam(Volume ~ s(Height) + s(Girth),
                  family=Gamma(link=log), data=trees)
```

  where the log function is just for stability.

- When we type `tree.gam.1` we get:

```
Family: Gamma
Link function: log

Formula:
Volume ~ s(Height) + s(Girth)

Estimated degrees of freedom:                GCV score: 0.0080824
1.0000 2.4222  total = 4.4223
```

- The EDFs are for: `Height`, `Girth`, and the total is the sum of these plus one for the intercept.

- EDF of one means essentially a straight line and therefore not worth smoothing.

Botanical Example

- So use:

```
summary(tree.gam.1)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.276      0.015     219   <2e-16

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(Height) 1.00   1.00  31.2 6.7e-06
s(Girth)  2.42   2.42 268.9 < 2e-16

R-sq.(adj) =  0.973   Deviance explained = 97.8%
GCV score = 0.0080824  Scale est. = 0.0069294  n = 31
```

What These Quantities Mean

- The standard intercept term:

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.276      0.015     219   <2e-16
```

- The smoothed terms:

```
Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(Height) 1.00   1.00  31.2 6.7e-06
s(Girth)  2.42   2.42 268.9 < 2e-16
```

where:

- ▻ `edf` gives the effective degrees of freedom (the trace of the $\mathbf{A}$ matrix). `1.00` means essentially a straight line.
- ▻ `Ref.df`, uses an alternative estimate of edf. Useful for testing
- ▻ `F p-value` give a Wald test of $\beta_j = 0$.

What These Quantities Mean

- `R-sq.(adj) = 0.973`, approximately the square of the correlation between observed and fitted values, adjusted for the degrees of freedom.
- `Deviance explained = 97.8`, model deviance in the GLM sense (not penalized deviance).
- `GCV score = 0.0080824`, minimized GCV score.
- `Scale est.  = 0.0069294`, estimated (or given) scale parameter σ^2.
- `n = 31`, data size without any adjustment.

Some Other Quantities of Interest

```
tree.gam.1$null.deviance
[1] 8.32

tree.gam.1$df.residual
[1] 26.6

tree.gam.1$hat
 [1] 0.2909 0.2502 0.2513 0.0744 0.1279 0.1613 0.1458 0.0615 0.0961 0.0590
[11] 0.0796 0.0593 0.0593 0.1056 0.0596 0.0727 0.1611 0.1837 0.1124 0.2674
[21] 0.0822 0.0961 0.0965 0.1443 0.1052 0.1148 0.1222 0.1301 0.1350 0.1350
[31] 0.5818
```

Formal Model Comparison

```
tree.gam.0 <- gam(Volume ~ s(Height),
                  family=Gamma(link=log), data=trees)

anova(tree.gam.0, tree.gam.1, test = "F")

Analysis of Deviance Table

Model 1: Volume ~ s(Height)
Model 2: Volume ~ s(Height) + s(Girth)
  Resid. Df Resid. Dev   Df Deviance   F Pr(>F)
1      29.0       4.77
2      26.6       0.18 2.42     4.59 273 <2e-16
```

Botanical Example

- There are also other quantities in the output:

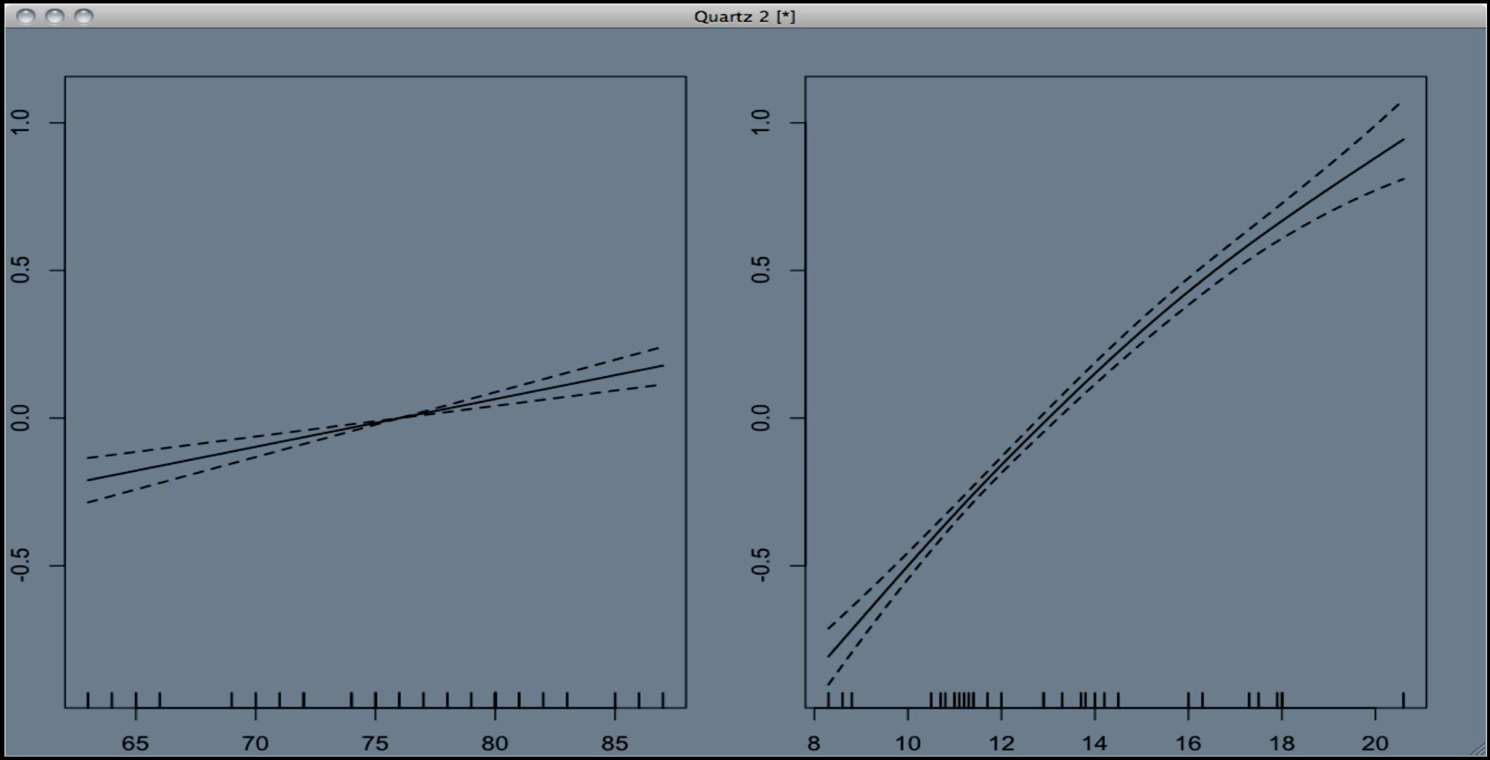
```
 names(tree.gam.1)
  [1] "coefficients"      "residuals"         "fitted.values"     "family"
  [6] "deviance"          "null.deviance"     "iter"              "weights"
 [11] "df.null"           "y"                 "converged"         "boundary"
 [16] "reml.scale"        "aic"               "rank"              "K"
 [21] "gcv.ubre"          "outer.info"        "scale"             "Vp"
 [26] "Ve"                "edf"               "nsdf"              "sig2"
 [31] "method"            "smooth"            "formula"           "var.summary"
 [36] "model"             "control"           "terms"             "pterms"
 [41] "offset"            "df.residual"       "min.edf"           "optimizer"
```

  but most of these you do not need to use.

- Graphing GAM output always helps:

```
postscript("Class.Stat.Comp/tree.fig1.ps")
par(mfrow=c(1,2),mar=c(4.5,4.5,2,2),cex.axis=1,cex.lab=1.1,bg="slategray")
plot(tree.gam.1,lwd=1.5)
dev.off()
```

Botanical Example



Botanical Example

- We used the default smoother: thin plate regression splines, order of penalty equal to two and the dimension of the basis equal to 10.
- Now change this to a penalized cubic regression spline for Girth:

```
tree.gam.2 <- gam(Volume ~ s(Height) + s(Girth,bs="cr",k=20),
                  family=Gamma(link=log), data=trees)
summary(tree.gam.2)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.276      0.015     219   <2e-16

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(Height) 1.00   1.00  31.2 6.7e-06
s(Girth)  2.42   2.42 266.4 < 2e-16

R-sq.(adj) =  0.973   Deviance explained = 97.8%
GCV score = 0.008083  Scale est. = 0.0069294  n = 31
```

Botanical Example

- A parameter, γ, is used to adjust the fit by multiplying the effective degrees of freedom.
- The default value for γ is 1, and higher values give smoother fits.
- Sometimes GCV gives overly rough fits (to some tastes), so Kim & Gu suggest using 1.4:

```
tree.gam.3 <- gam(Volume ~ s(Height) + s(Girth,bs="cr",k=20),
                  family=Gamma(link=log), data=trees, gamma=1.4)
summary(tree.gam.3)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.2758     0.0151     218   <2e-16

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(Height) 1.00   1.00  31.4 6.1e-06
s(Girth)  2.17   2.17 294.2 < 2e-16

R-sq.(adj) =  0.972   Deviance explained = 97.7%
GCV score = 0.0092292  Scale est. = 0.0070251  n = 31
```

Details on GAM Model Specification

- The `R` formula for `gam` is just like `glm` except we have new smoother terms: `s` and `te`.
- The notation `s(X1)`, gives a spline based smooth for the `X1` explanatory variable.
- The notation `te(X2)` gives a tensor product based smooth for `X2` explanatory variable.
- It is common to "mix" smoothed and unsmoothed terms in a model:

```
Y ~ X1 + s(X2) + te(X3)
```

- There can be nested smoothing specifications:

```
Y ~ s(X1) + s(X2) + s(X1,X2)
Y ~ s(X1,X2) + s(X2,X3)
```

- We can also control the smooth with parameter vectors, for instance:

```
Y ~ te(X1,X2, bs=c("tp","tp"), m=c(3,4), k=(5,6))
```

which gives a tensor product smooths of `X1` and `X2` with bases of dimension 3 for `X1` and 4 for `X2`, and marginal penalties of 5 for `X1` and 6 for `X2`.

Chronic Bronchitis and Dust Concentration Study

- The file contains data from a study of the Deutsche Forschungsgemeinschaft. The data were recorded during the years 1960 and 1977 in a Munich plant (1246 workers).

- Objective: dose response model for cbr with covariates dust, expo and smoking, and assessment of threshold limiting value under which dust has no influence on cbr.

- Description of the variables:

cbr	Chronic Bronchitis Reaction
	1 : Yes
	0 : No
dust	dust concentration at working place (in mg/m)
smoking	does worker smoke?
	1 : Yes
	0 : No
expo	duration of exposure in years

Chronic Bronchitis and Dust Concentration Study

- Sources:
  Gossl, C. / Kuchenhoff, H. (2001): Bayesian analysis of logistic regression with an unknown change point and covariate measurement error. Statistics in Medicine, 20, 3109-3121.
  Kuchenhoff, H. / Carroll, R.J. (1997): Segmented regression with errors in predictors: semiparametric and parametric methods. Statistics in Medicine, 16, 169-188.

Chronic Bronchitis and Dust Concentration Study, Read Data and Run a GLM

dust.df ¡- read.table(”http://jgill.wustl.edu/data/dust.asc”,header=TRUE)

```
dust.df <- read.table("/Users/jgill/Class.GLM/dust.asc", header=TRUE)
summary(dust.df)

      cbr                dust               smoking              expo
 Min.   :0.0000   Min.    : 0.2000   Min.   :0.0000   Min.   : 3.00
 1st Qu.:0.0000   1st Qu.: 0.4925   1st Qu.:0.0000   1st Qu.:16.00
 Median :0.0000   Median : 1.4050   Median :1.0000   Median :25.00
 Mean   :0.2343   Mean   : 2.8154   Mean   :0.7392   Mean   :25.06
 3rd Qu.:0.0000   3rd Qu.: 5.2475   3rd Qu.:1.0000   3rd Qu.:33.00
 Max.   :1.0000   Max.   :24.0000   Max.   :1.0000   Max.   :66.00

dust.glm <- glm(cbr ~ dust+smoking+expo, family = binomial(link = logit),
        data=dust.df);
```

Chronic Bronchitis and Dust Concentration Study, GLM Results

```
summary.glm(dust.glm)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3675  -0.7798  -0.5906  -0.3813   2.3022

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.04785    0.24813 -12.283  < 2e-16
dust         0.09189    0.02323   3.956 7.63e-05
smoking      0.67683    0.17407   3.888 0.000101
expo         0.04016    0.00620   6.476 9.40e-11
---
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1356.8  on 1245  degrees of freedom
Residual deviance: 1278.3  on 1242  degrees of freedom
AIC: 1286.3
```

Chronic Bronchitis and Dust Concentration Study, Run a GAM

```
library(mgcv)       # DOWNLOAD FROM CRAN IF NECESSARY, INCLUDES TENSOR SMOOTH
                    # WRITTEN BY SIMON WOOD
# library(gam)      # TREVOR HASTIE'S OLDER PACKAGE
```

- To get started we will just use the default smoother: thin place regression splines, order of penalty equal to two.

```
dust.gam <- gam(cbr ~ s(dust,k=32) + smoking + s(expo,k=32),
                family = binomial(link = logit), data=dust.df)
```

- Note the use of `s()` here to denote "spline"
- Here 32 is the dimension of the basis used to represent the smooth term in both cases.
- The GAM penalized likelihood maximization problem is solved by *Penalized Iteratively Reweighted Least Squares*.

Chronic Bronchitis and Dust Concentration Study, GAM Results

```
summary(dust.gam)
Family: binomial
Link function: logit

Formula:
cbr ~ s(dust, k = 32) + smoking + s(expo, k = 32)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.410      1.555   -1.55  0.12114
smoking        0.673      0.179    3.77  0.00016

Approximate significance of smooth terms:
         edf Ref.df Chi.sq p-value
s(dust) 24.3   24.3   31.7    0.14
s(expo)  3.9    3.9   54.4 3.7e-11

R-sq.(adj) =  0.121   Deviance explained = 13.3%
UBRE score = -0.0074567  Scale est. = 1         n = 1246
```

Chronic Bronchitis and Dust Concentration Study, Explaining GAM OUTPUT

- `Parametric coefficients`: read like normal GLM output.
- `edf`: coefficient's estimated degrees of freedom (penalization means that many of these are less than 1)
- `Ref.df`: the same for us. Note also:

```
dust.gam$df.residual
[1] 1215.768
dust.gam$df.null
[1] 1245
sum(dust.gam$edf)
[1] 30.23168
dust.gam$min.edf
[1] 6
```

- `R-sq.(adj)`: no need to pay attention to this
- `Deviance explained`: equivalent to $(D_n - D_m)/D_n$, i.e.:

```
1-dust.gam$deviance/dust.gam$null.deviance
[1] 0.1331007
```

Chronic Bronchitis and Dust Concentration Study, Explaining GAM OUTPUT

- `UBRE score`: the UnBiased Risk Estimator estimated by $D/n + 2s(DoF)/(n - s)$, where D is the deviance, n is the number of cases, s the scale parameter and DoF is the effective degrees of freedom of the model. UBRE is the AIC only rescaled, and should be used only when s is known.
- Using `dust.gam$aic` we could get the AIC but it is misleading since we maximize the penalized likelihood rather than the regular likelihood, and these have different degrees of freedom.

Contrasting GLM and GAM Output

- Results from the GLM:

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.04785    0.24813 -12.283  < 2e-16
dust         0.09189    0.02323   3.956 7.63e-05
smoking      0.67683    0.17407   3.888 0.000101
expo         0.04016    0.00620   6.476 9.40e-11
```

- Results from the GAM:

```
Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.410      1.555   -1.55  0.12114
smoking        0.673      0.179    3.77  0.00016

Approximate significance of smooth terms:
         edf Ref.df Chi.sq p-value
s(dust) 24.3   24.3   31.7    0.14
s(expo)  3.9    3.9   54.4 3.7e-11
```

Predictions From GAM Output

- First build a design matrix for non-smokers that has values over the range of `dust` and `expo` from their maximum to their minimum.

```
attach(dust.df);      logit <- function(Xb)  1/(1+exp(-Xb))
( predict.dust.df <- data.frame(dust=seq(min(dust),max(dust),length=20),
     smoking=rep(0,length=20),expo=seq(min(expo),max(expo),length=20)) )

predict.dust.df
        dust smoking      expo
1    0.200000       0  3.000000
2    1.452632       0  6.315789
3    2.705263       0  9.631579
4    3.957895       0 12.947368
5    5.210526       0 16.263158
6    6.463158       0 19.578947
7    7.715789       0 22.894737
8    8.968421       0 26.210526
9   10.221053       0 29.526316
10  11.473684       0 32.842105
:   :               : :
```

Predictions From GAM Output

```
:  :                :  :
11 12.726316        0 36.157895
12 13.978947        0 39.473684
13 15.231579        0 42.789474
14 16.484211        0 46.105263
15 17.736842        0 49.421053
16 18.989474        0 52.736842
17 20.242105        0 56.052632
18 21.494737        0 59.368421
19 22.747368        0 62.684211
20 24.000000        0 66.000000

predict.dust.dens <- matrix(NA,20,20)
for (i in 1:20)  {
    predict.dust.df.temp <- data.frame(dust=rep(predict.dust.df$dust[i],length=20),
        smoking=rep(0,length=20),expo=seq(min(expo),max(expo),length=20))
    predict.dust.dens[i,] <-
        logit(predict.gam(dust.gam,newdata=predict.dust.df.temp,se.fit=F,plot.call=F))
}
```

Predictions From GAM Output

- Now build a design matrix for smokers that has also values over the range of `dust` and `expo` from their maximum to their minimum.

```
( predict.dust.df2 <- data.frame(dust=seq(min(dust),max(dust),length=20),
    smoking=rep(1,length=20),expo=seq(min(expo),max(expo),length=20)) )

predict.dust.df2
        dust smoking      expo
1   0.200000       1  3.000000
2   1.452632       1  6.315789
3   2.705263       1  9.631579
4   3.957895       1 12.947368
5   5.210526       1 16.263158
6   6.463158       1 19.578947
7   7.715789       1 22.894737
8   8.968421       1 26.210526
9  10.221053       1 29.526316
10 11.473684       1 32.842105
:  :                 : :
```

Predictions From GAM Output

```
:  :                  : :
11 12.726316          1 36.157895
12 13.978947          1 39.473684
13 15.231579          1 42.789474
14 16.484211          1 46.105263
15 17.736842          1 49.421053
16 18.989474          1 52.736842
17 20.242105          1 56.052632
18 21.494737          1 59.368421
19 22.747368          1 62.684211
20 24.000000          1 66.000000

predict.dust.dens2 <- matrix(NA,20,20)
for (i in 1:20)  {
    predict.dust.df2.temp <- data.frame(dust=rep(predict.dust.df2$dust[i],length=20),
        smoking=rep(1,length=20),expo=seq(min(expo),max(expo),length=20))
    predict.dust.dens2[i,] <-
        logit(predict.gam(dust.gam,newdata=predict.dust.df2.temp,se.fit=F,plot.call=F))
}
```

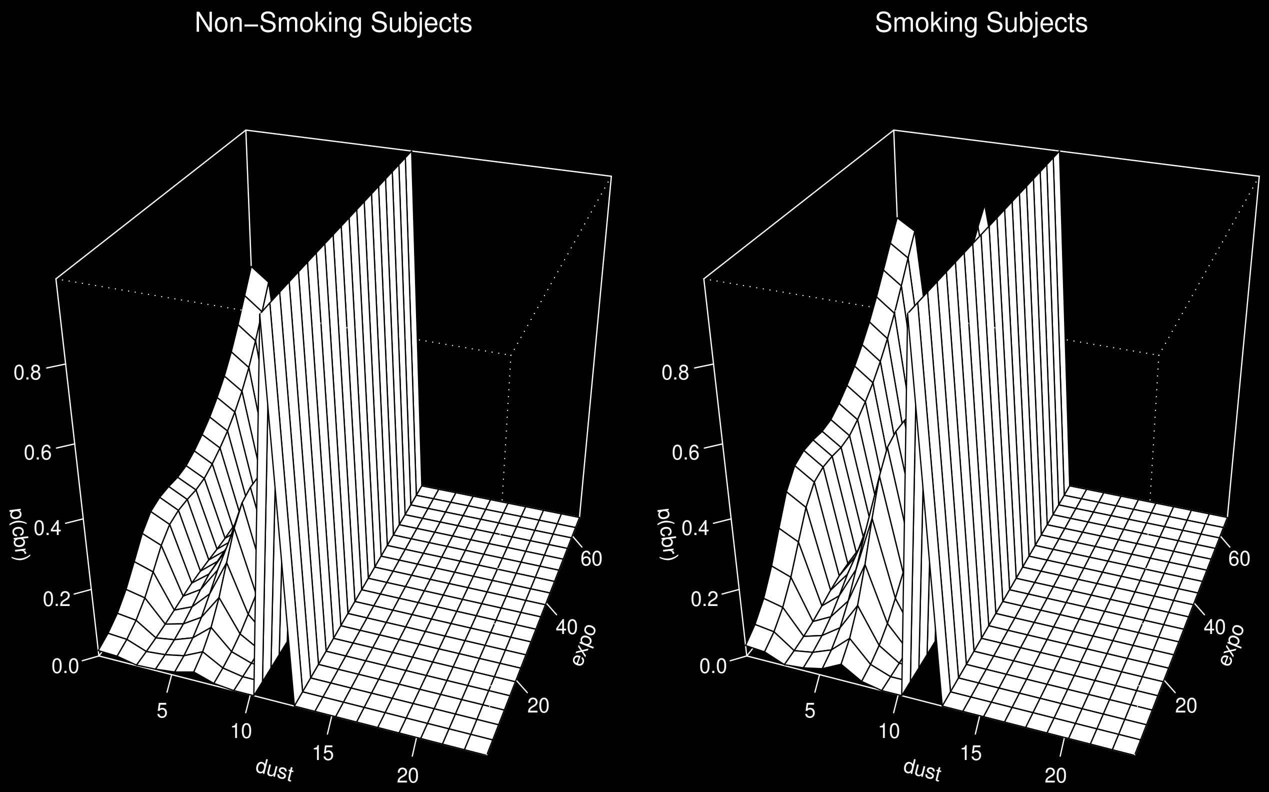
Predictions From GAM Output, Perspective Plot

- Graph with a perspective plot...
- The surface is then viewed by looking at the origin from a direction defined by `theta` and `phi`.
- If `theta` and `phi` are both zero the viewing direction is directly down the negative y axis.
- Changing `theta` will vary the *azimuth* and changing `phi` the *colatitude*.
- The term `r` is the distance of the eyepoint from the center of the plotting box.

Predictions From GAM Output, Perspective Plot

```
postscript("Class.Stat.Comp/dust.gam1.ps")
par(mfrow=c(1,2),mar=c(1,1,1,1),oma=c(1,1,1,1),col.axis="white",col.lab="black",
    col.sub="white",col="white",bg="black")

persp(predict.dust.df$dust,predict.dust.df$expo,predict.dust.dens,
    theta=20,phi=30,r=5,ticktype="detailed",xlab="dust",ylab="expo",zlab="p(cbr)")

mtext(side=3,outer=F,cex=1.3,"Non-Smoking Subjects",line=-1)

persp(predict.dust.df2$dust,predict.dust.df2$expo,predict.dust.dens2,
    theta=20,phi=30,r=5,ticktype="detailed",xlab="dust",ylab="expo",zlab="p(cbr)")

mtext(side=3,outer=F,cex=1.3,"Smoking Subjects",line=-1)

dev.off()
```

Predictions From GAM Output, Perspective Plot



Predictions From GAM Output, Plot Like McCullagh and Nelder

```
library(modreg)

# NON-SMOKERS, X-AXIS IS DUST
attach(dust.df)
postscript("Class.Stat.Comp/dust.gam3.ps")
par(mfrow=c(1,2),mar=c(3,3,3,3),oma=c(1,1,1,1),col.axis="white",col.lab="black",
    col.sub="white",col="white",bg="black")
plot(dust,dust.gam$fitted.values,pch=".",ylab="fitted values")
    predict.dust.df.temp <- data.frame(dust=predict.dust.df$dust,
                    smoking=rep(0,length=20),expo=rep(mean(expo),length=20))
    predict.fit <- predict.gam(dust.gam,newdata=predict.dust.df.temp,se.fit=T)
    mu.fit<-logit(predict.fit$fit)
    lines(predict.dust.df$dust,mu.fit,lwd=2)
    lines(predict.dust.df$dust,logit(predict.fit$fit-predict.fit$se.fit))
    lines(predict.dust.df$dust,logit(predict.fit$fit+predict.fit$se.fit))
    mtext(outer=F,side=3,line=1,"Non-smokers")
```

Predictions From GAM Output, Plot Like McCullagh and Nelder

```
# SMOKERS, X-AXIS IS DUST
plot(dust,dust.gam$fitted.values,pch=".",ylab="fitted values")
    predict.dust.df.temp <- data.frame(dust=predict.dust.df$dust,
                    smoking=rep(1,length=20),expo=rep(mean(expo),length=20))
    predict.fit <- predict.gam(dust.gam,newdata=predict.dust.df.temp,se.fit=T)
    mu.fit<-logit(predict.fit$fit)
    lines(predict.dust.df$dust,mu.fit,lwd=2)
    lines(predict.dust.df$dust,logit(predict.fit$fit-predict.fit$se.fit))
    lines(predict.dust.df$dust,logit(predict.fit$fit+predict.fit$se.fit))
    mtext(outer=F,side=3,line=1,"Smokers")

dev.off()
detach(dust.df)
```

Full Syntax for `gam`

- There are many modeling options:

```
gam(formula, family=gaussian(), data=list(), weights=NULL,
    subset=NULL, na.action, offset=NULL, method="GCV.Cp",
    optimizer=c("outer","newton"), control=gam.control(),
    scale=0, select=FALSE, knots=NULL, sp=NULL,min.sp=NULL,
    H=NULL,gamma=1, fit=TRUE, paraPen=NULL,G=NULL, in.out,...)
```

with:

`formula`	a full `R` modeling formula, including smooth terms
`family`	if "gaussian" fitting is by least-squares, and if "symmetric" by a re-descending M-estimator
`data`	an optional data frame, list or environment
`weights`	optional regression-style weights for each case
`subset`	an optional subset of the data to be used
`na.action`	the regular model treatment of missing data
`offset`	used to supply a model offset for use in fitting
`control`	control parameters, see `gam.control`
`method`	smoothing parameter estimation method "GCV.Cp" to use GCV for unknown scale parameter and Mallows' Cp/UBRE/AIC for known scale. "GACV.Cp" is equivalent, but using GACV in place of GCV. "REML" for REML estimation, including of unknown scale, "P-REML" for REML estimation, but using a Pearson estimate of the scale. "ML" and "P-ML" are similar, but using maximum likelihood in place of REML
`optimizer`	"perf" for performance iteration, "outer" for the more stable direct approach. "outer" can use several alternative optimizers, specified in the second element of optimizer: "newton" (default), "bfgs", "optim", "nlm" and "nlm.fd" (slow)
`scale`	positive values for the scale parameter, negative for unknown, zero for 1 into Poisson and binomial and unknown for other distributions

Full Syntax for `gam`

`select`	If `TRUE` then the fit can an extra penalty to each term penalized towards zero
`knots`	list containing user specified knot values (must match k value supplied
`sp`	smoothing parameter vector in the order that the smooth terms appear in the model formula, negative elements indicate that the parameter should be estimated
`min.sp`	lower bounds for smoothing parameters
`H`	user supplied fixed quadratic penalty on the parameters, often for ridge
`gamma`	multiplier to inflate the model degrees of freedom in the GCV or UBRE/AIC score
`fit`	If `TRUE` then model is fit, if `FALSE` then the model is set up and an object G containing what would be required to fit is returned is returned
`paraPen`	optional list specifying any penalties to be applied to parametric model terms
`G`	object returned by a previous call to gam with `fit=FALSE`
`in.out`	optional list for initializing outer iteration

Terrorism Data Analysis

- This example is about comparing different GAM fits.
- Source: The International Policy Institute for Counter-Terrorism, Herzlia, Israel.
- Provided on an online database with details of attacks in Israel since September, 2000.
- Subsetted by Markison to give 103 suicide attacks over a three-year period from November 6, 2000 to November 3, 2003 when there was a steep drop.
- Information provided: date and place of the attack, attack type, the type of target and device employed, organizational affiliation of the attacker, and the number of casualties, along with a written description of the attack.
- Casualties are given personal attributes such as name, age, sex, nationality, and religion.

ison3.txt",header=TRUE)

Terrorism Data

```
harr <- read.table("http://jgill.wustl.edu/data/harrison4.txt",header=TRUE)
apply(harr[,-1],2,table)

$NumberKilled
 0  1  2  3  5  6  7  8  9 11 15 17 19 21 23 24 30
44 13  9  8  3  2  3  2  2  3  4  3  1  3  1  1  1

$NumberInjured
  0   1   2   3   4   5   6   8   9  11  13  14  16  17  20  21  22  26  27  30
 28   1   5   4   4   3   1   2   2   2   1   1   1   1   3   1   1   1   1   5
 40  42  47  50  52  57  58  59  60  65  69  86  90 100 102 120 130 150 188
  3   1   1   7   1   1   1   2   5   1   1   1   1   3   1   1   1   2   1

$TotalCasualties
  0   1   2   3   4   5   6   8   9  10  12  13  15  17  20  21  26  27  29  30
 22   5   6   4   3   3   2   2   1   1   2   2   1   1   2   1   1   2   1   1
 31  32  35  38  45  49  50  51  52  53  57  58  59  61  62  63  65  67  71  75
  1   1   1   1   1   2   1   1   1   1   2   1   1   2   1   1   2   2   3   2
 81  91  93 105 106 123 126 141 145 151 180 199
  1   1   1   1   1   1   1   1   1   1   1   1
```

Terrorism Data

```
$ResponsibleHamas
 0  1
59 44

$ResponsibleisPIJ
 0  1
79 24

$TargetisMilitary
 0  1
76 10

$TargetisBus
 0  1
89 14

$TargetisCheckpoint
 0  1
87 16
```

```
$ResponsibleisMartyrs
 0  1
78 25

$ResponsibleisOther
 0  1
99  4

$TargetisCivilian
 0  1
10 76

$TargetisCafe
 0  1
89 14

$TargetisResidence
  0   1
102   1
```

Terrorism Data

```
$TargetisOffshore
  0   1
101   2

$TargetisStreet
 0  1
71 32

$DeviceisCar
 0  1
89 14

$AttackisPrevented
  0   1
101   2

$FirstAttackerisMale
 0  1
 7 92
```

```
$TargetisStore
  0  1
 96  7

$TargetisTravelstop
  0  1
 88 15

$DeviceisBoat
  0   1
101   2

$AttackerisChallenged
 0  1
63 40

$FirstAttackerisFemale
 0  1
92  7
```

Terrorism Data

```
$AgeofFirstAttacker
16 17 18 19 20 21 22 23 24 25 26 27 29 31 43 45 48
 1  8  7 10 15 11 10 12  2  3  2  1  3  1  1  1  1
```

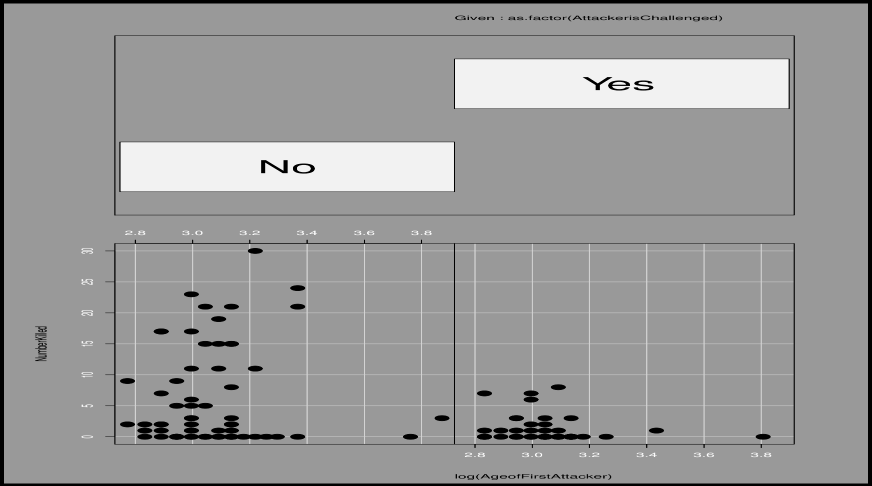
- Data Notes:
  - measurement is very nongranular,
  - some dichotomous variables very skewed,
  - and real motivations, planning, and training are not observed.

Terrorism Data Analysis

```
postscript("Class.Stat.Comp/coplot1.ps")
par(mfrow=c(1,1),mar=c(6,6,6,2),col.axis="white",col.lab="white", col.sub="white",col="black",bg="grey60", cex.lab=.001)
coplot2(NumberKilled ~ log(AgeofFirstAttacker) | as.factor(AttackerisChallenged), data = harr, cex=2, pch=19)
mtext(side=1,line=10,"log(Age of First Attacker)",cex=2)
mtext(side=2,line=10,"Number Killed",cex=2)
mtext(side=3,line=10,"Attacker Is Challenged",cex=2)
dev.off()

postscript("Class.Stat.Comp/coplot2.ps")
par(mfrow=c(1,1),mar=c(6,6,6,2),col.axis="white",col.lab="white", col.sub="white",col="black",bg="grey60", cex.lab=.001)
coplot2(NumberKilled ~ log(AgeofFirstAttacker) | as.factor(DeviceisCar), data = harr,cex=2,pch=19)
mtext(side=1,line=10,"log(Age of First Attacker)",cex=2)
mtext(side=2,line=10,"Number Killed",cex=2)
mtext(side=3,line=10,"Attacker Is Challenged",cex=2)
dev.off()

postscript("Class.Stat.Comp/coplot3.ps")
par(mfrow=c(1,1),mar=c(6,6,6,2),col.axis="white",col.lab="white", col.sub="white",col="black",bg="grey60", cex.lab=.001)
coplot2(NumberKilled ~ log(AgeofFirstAttacker) | as.factor(TargetisMilitary), data = harr,cex=2,pch=19)
mtext(side=1,line=10,"log(Age of First Attacker)",cex=2)
mtext(side=2,line=10,"Number Killed",cex=2)
mtext(side=3,line=10,"Attacker Is Challenged",cex=2)
dev.off()

postscript("Class.Stat.Comp/coplot4.ps")
par(mfrow=c(1,1),mar=c(6,6,6,2),col.axis="white",col.lab="white", col.sub="white",col="black",bg="grey60", cex.lab=.001)
coplot2(NumberKilled ~ log(AgeofFirstAttacker) | as.factor(ResponsibleHamas), data = harr,cex=2,pch=19)
mtext(side=1,line=10,"log(Age of First Attacker)",cex=2)
mtext(side=2,line=10,"Number Killed",cex=2)
mtext(side=3,line=10,"Attacker Is Challenged",cex=2)
dev.off()
```

Terrorism Data Analysis

Attacker is Challenged



Device is Car

Target is Military

Hamas Responsible

Terrorism Data Analysis

```
harr.gam1 <- gam(NumberKilled ~ s(log(AgeofFirstAttacker),bs="tp") + log(Date) +
                AttackerisChallenged + FirstAttackerisFemale +
                DeviceisCar + TargetisCafe + TargetisMilitary +
                ResponsibleHamas, data=harr)

harr.gam2 <- gam(NumberKilled ~ s(log(AgeofFirstAttacker),bs="tp") +
                s(log(Date),bs="cr",k=5) +
                AttackerisChallenged + FirstAttackerisFemale +
                DeviceisCar + TargetisCafe + TargetisMilitary +
                ResponsibleHamas, data=harr)
```

Terrorism Data Analysis

```
summary(harr.gam1)

Parametric coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            -5.1905     2.4608  -2.109 0.037602
log(Date)               2.5481     0.6210   4.103 8.72e-05
AttackerisChallenged   -3.7087     1.1381  -3.259 0.001563
FirstAttackerisFemale   2.0103     2.1901   0.918 0.361043
DeviceisCar             0.8684     1.6705   0.520 0.604415
TargetisCafe            4.0685     1.6358   2.487 0.014656
TargetisMilitary       -4.5712     1.4032  -3.258 0.001568
ResponsibleHamas        4.0489     1.1692   3.463 0.000809

Approximate significance of smooth terms:
                              edf Ref.df     F p-value
s(log(AgeofFirstAttacker)) 1.893  1.893 1.159   0.316

R-sq.(adj) =  0.391   Deviance explained = 44.4%
GCV score = 30.657  Scale est. = 27.712    n = 103
```

Terrorism Data Analysis

```
summary(harr.gam2)

Parametric coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)              3.879      1.065   3.643 0.000446
AttackerisChallenged    -3.673      1.135  -3.237 0.001684
FirstAttackerisFemale    2.096      2.185   0.959 0.339952
DeviceisCar              1.185      1.699   0.698 0.487152
TargetisCafe             4.100      1.627   2.520 0.013481
TargetisMilitary        -4.599      1.402  -3.280 0.001467
ResponsibleHamas         4.544      1.225   3.709 0.000356

Approximate significance of smooth terms:
                            edf Ref.df     F  p-value
s(log(AgeofFirstAttacker)) 1.811  1.811 1.465 0.236896
s(log(Date))               2.503  2.503 6.978 0.000633

R-sq.(adj) =  0.396   Deviance explained = 45.7%
GCV score = 30.911  Scale est. = 27.515    n = 103
```

Terrorism Data Analysis

- It is also possible to do simultaneous multivariate smoothing:

```
harr.gam3 <- gam(NumberKilled ~ te(log(AgeofFirstAttacker),log(Date),k=3) +
                AttackerisChallenged + FirstAttackerisFemale +
                DeviceisCar + TargetisCafe + TargetisMilitary +
                ResponsibleHamas, data=harr)
```

- This fits a bivariate surface for `log(AgeofFirstAttacker)` and `log(Date)` at the same time using a *tensor product smooth*.

- In this case it is a slightly better fit. . .

Terrorism Data Analysis

```
summary(harr.gam3)

Parametric coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)              4.377      1.070   4.091  9.3e-05
AttackerisChallenged    -4.059      1.144  -3.549 0.000616
FirstAttackerisFemale    1.286      2.255   0.571 0.569737
DeviceisCar              1.204      1.677   0.718 0.474763
TargetisCafe             3.824      1.638   2.335 0.021752
TargetisMilitary        -4.772      1.384  -3.448 0.000860
ResponsibleHamas         4.027      1.184   3.400 0.001003

Approximate significance of smooth terms:
                                        edf Ref.df     F p-value
te(log(AgeofFirstAttacker),log(Date)) 5.613  5.613 3.794 0.00255

R-sq.(adj) =  0.412   Deviance explained = 47.9%
GCV score = 30.527  Scale est. = 26.789    n = 103
```

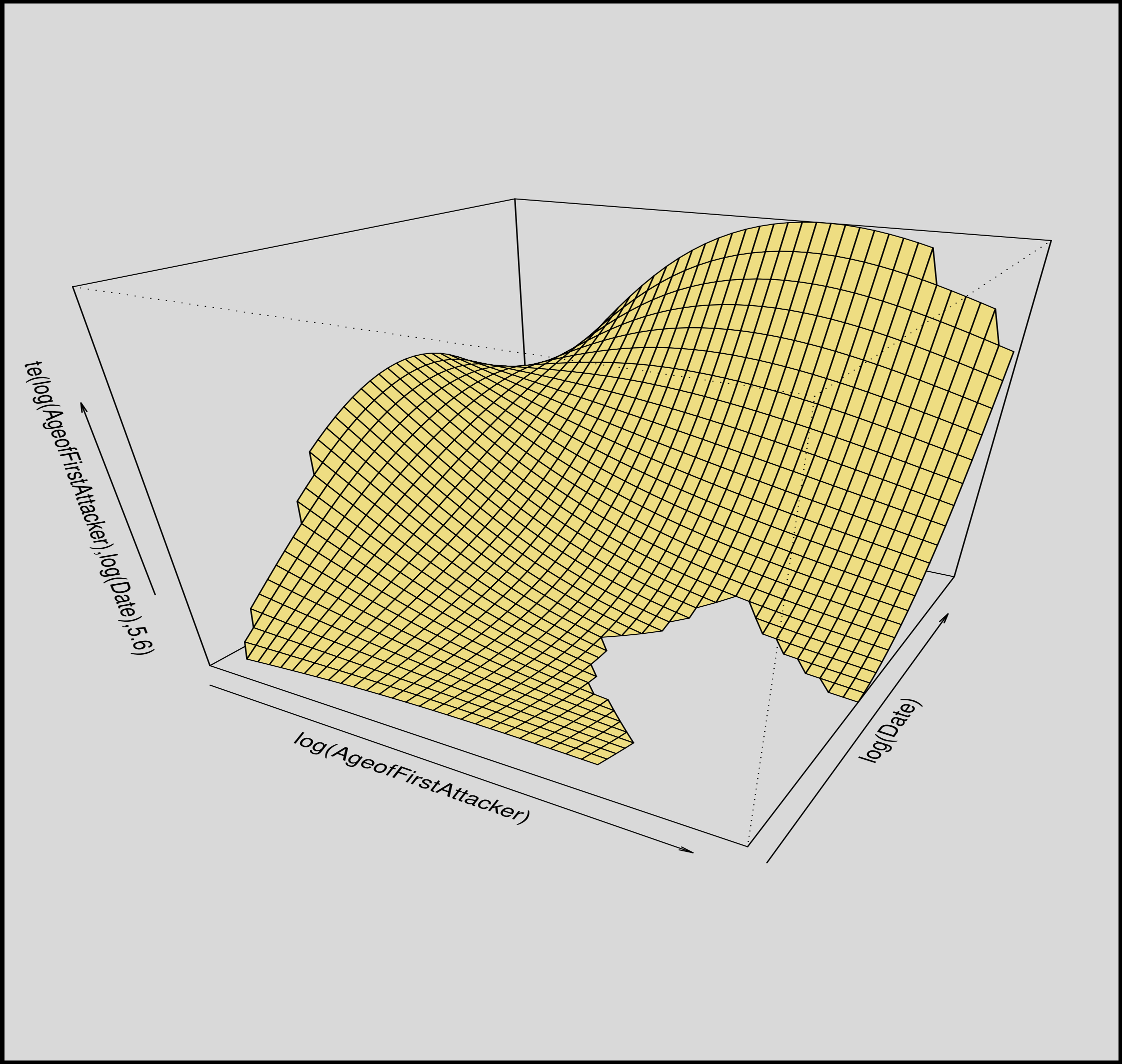
Graphing the Terrorism Data Analysis

- There is a handy but slightly confusing plot routine for GAMs:

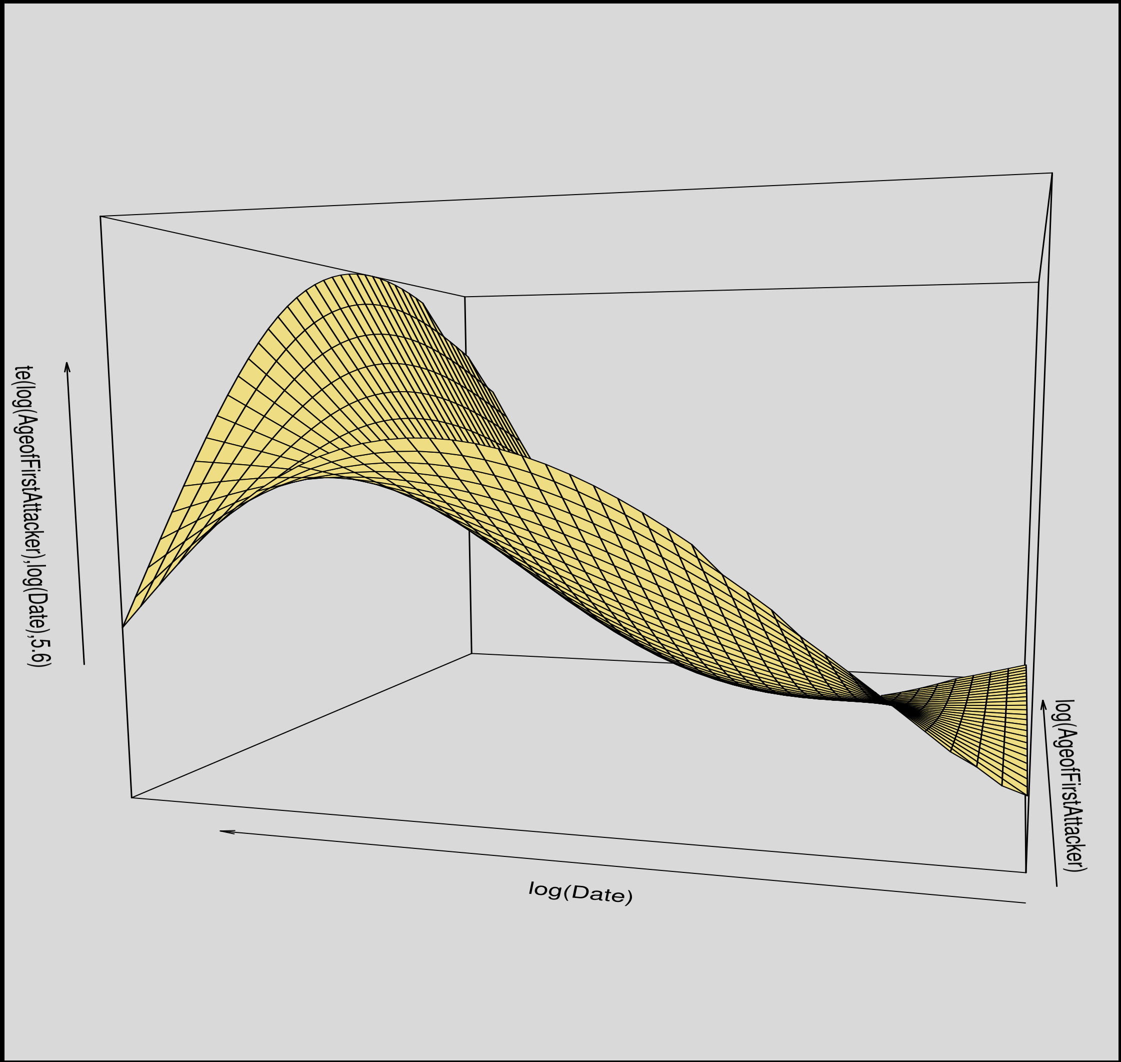
```
postscript("Class.Stat.Comp/harr.persp1.ps")
par(mfrow=c(1,1),mar=c(1,1,0,1),oma=c(0,0,0,0),col.axis="white",col.lab="white",
     col.sub="white",col="black",bg="grey85",cex.lab=3)
plot.gam(harr.gam3,too.far=0.25,lwd=1,pers=TRUE,col="lightgoldenrod",cex.lab=1.3)
dev.off()
postscript("Class.Stat.Comp/harr.persp2.ps")
par(mfrow=c(1,1),mar=c(1,2,0,1),oma=c(0,0,0,0),col.axis="white",col.lab="white",
     col.sub="white",col="black",bg="grey85",cex.lab=3)
plot.gam(harr.gam3,too.far=0.25,lwd=1,pers=TRUE,col="lightgoldenrod",cex.lab=1.3,
         theta=285,phi=5)
dev.off()
```

- This option gives the perspective plot.

Viewing the Nonparametric Results



Viewing the Nonparametric Results

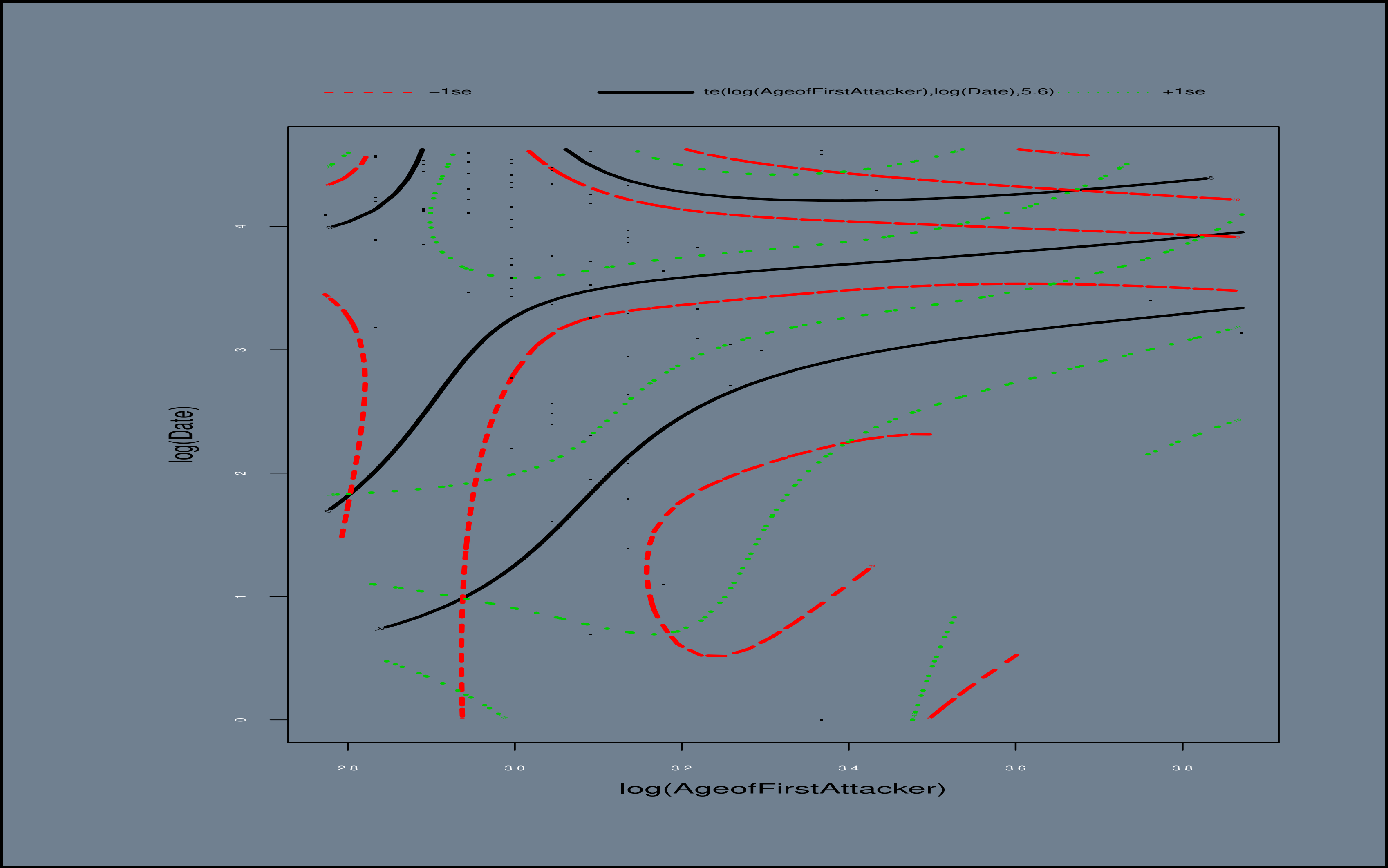


Graphing the Terrorism Data Analysis

- We can also plot the smooth function that results from the bivariate fit:

```
postscript("Class.Stat.Comp/harr.smooth.ps")
par(mfrow=c(1,1),mar=c(8,8,8,3),col.axis="white",col.lab="white",
    col.sub="white",col="black",bg="slategrey")
plot.gam(harr.gam3,too.far=0.25,lwd=3,cex.lab=2,cex.main=1.3)
dev.off()
```

Graphing the Terrorism Data Analysis

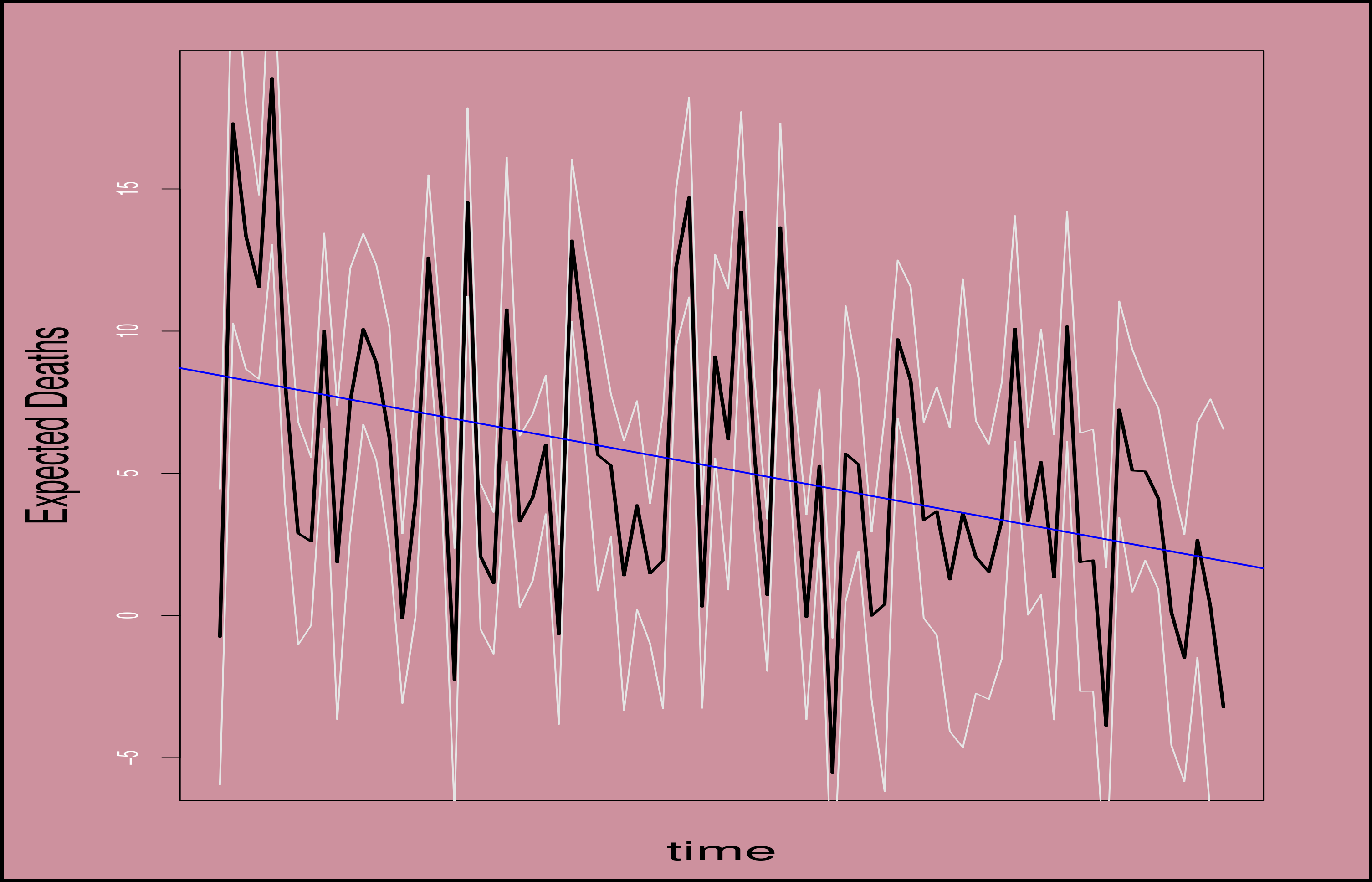


Graphing the Terrorism Data Analysis

- Recall that it is useful to see predictions on the outcome variable.
- Since our LHS is ordered by time, this provides a *serial prediction*.
- Simple process:

```
harr.gam3.predict <- predict(harr.gam3,se=TRUE,type="response")
postscript("Class.Stat.Comp/harr.predict.ps")
par(mfrow=c(1,1),mar=c(5,5,3,3),col.axis="white",col.lab="black",
    col.sub="white",col="black",bg="pink3",cex.lab=2)
plot(harr.gam3.predict$fit,lwd=2,type="l",ylab="Expected Deaths",
     xaxt="n",xlab="time")
lines(harr.gam3.predict$fit+1.96*harr.gam3.predict$se.fit,col="grey90")
lines(harr.gam3.predict$fit-1.96*harr.gam3.predict$se.fit,col="grey90")
y <- as.vector(harr.gam3.predict$fit); x <- 1:78
abline(lm(y ~ x),col="blue",lwd=2)
dev.off()
```

Graphing the Terrorism Data Analysis



GDP Growth

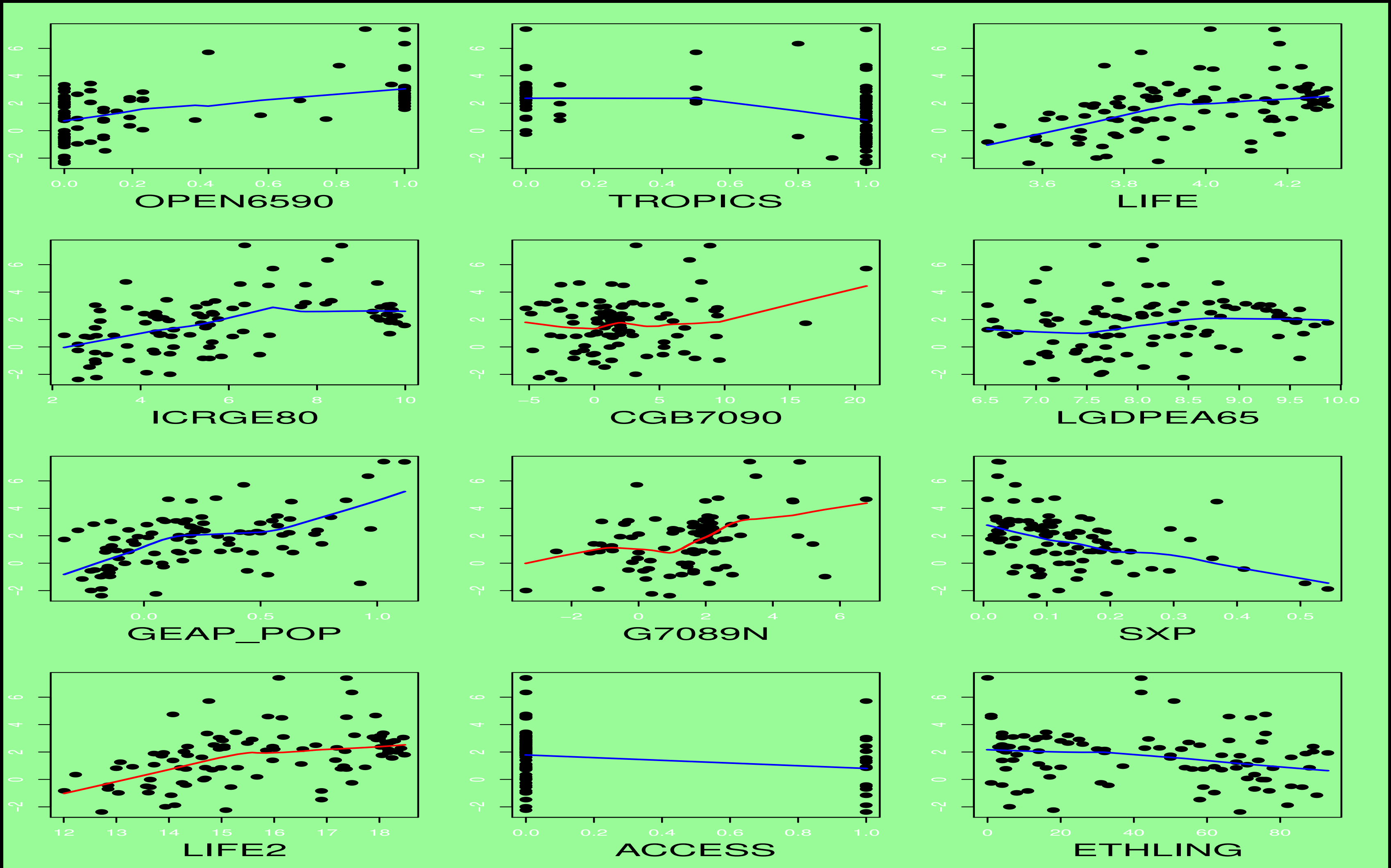
- This example is about how to tell if a GAM fit is better than an LM fit.
- Data used in Sachs and Warner (1997a, 1997b, 1995a, 1995b), eg. "Fundamental Source of Long-Run Growth", American Economic Review (1997b).
- See `http://www.bris.ac.uk/Depts/Economics/Growth/sachs.htm` for more details.
- Variables:
  - `GR6590` Average annual growth in real GDP per economically active population between 1970 and 1989.
  - `OPEN6590` The fraction of years during the period 1965-1990 in which the country is rated as an open economy.
  - `TROPICS` Takes the value 1 for a country in which the entire land area is subject to a tropical climate.
  - `LIFE` Log of life expectancy at birth, circa 1965-1970.
  - `ICRGE80` An average of 5 sub-indexes, each based on survey data from Political Risk Services.
  - `CGB7090` Central government savings is measured as current revenues minus current expenditures of the central government, expressed as a fraction of GDP.

- `LGDPEA65` Natural log of real (purchasing power parity adjusted) GDP per economically active population.
- `GEAP_POP` Difference between the growth rate of the economically active population (between ages 15 and 65) and growth of total population.
- `G7089N` Growth of neighboring countries.
- `INFL6590` Average inflation 1965-90.
- `SXP` Share of exports of primary products in GNP in 1970.
- `LIFE2` Life squared.
- `ACCESS` Physical access to international waters.
- `ETHLING` Ethno-linguistic fractionalization taken from related work by Mauro (1995) and Easterly and Levine (1996).

GDP Growth

```
sachs <- read.table("http://jgill.wustl.edu/data/sachs.csv",sep=",",header=TRUE)
library(mice)
imp.sachs <- mice(sachs[,c(3:15)],m=10)
comp.sachs <- complete(imp.sachs,1)
dim(comp.sachs)
95 13

postscript("Class.Stat.Comp/sachs1.ps")
par(mfrow=c(4,3),mar=c(5,2,2,2),col.axis="white",col.lab="black",
    col.sub="white",col="black",bg="palegreen",cex.lab=2)
for (i in c(1:9,11:13))  {
    plot(comp.sachs[,i],comp.sachs[,10],cex=1.25,pch=19,
        xlab=names(comp.sachs)[i],ylab="")
    lo.object <- lowess(comp.sachs[,10]~comp.sachs[,i],f=2/3)
    if (i==5 | i==8 | i==11) lines(lo.object$x,lo.object$y,lwd=2,col="red")
    else lines(lo.object$x,lo.object$y,lwd=2,col="blue")
}
dev.off()
```

GDP Growth, Graphed Against Explanatory Variables



GDP Growth

```
gdp.lm   <- lm(GR6590 ~ OPEN6590 + TROPICS + LIFE + ICRGE80 +
                CGB7090 + LGDPEA65 + GEAP_POP + G7089N +
                SXP + LIFE2 + ACCESS + ETHLING, data=comp.sachs)

gdp.gam1 <- gam(GR6590 ~ OPEN6590 + TROPICS + ICRGE80 +
                s(CGB7090,bs="cr",k=20) + LGDPEA65 + GEAP_POP +
                s(G7089N,bs="cr",k=20) + SXP + s(LIFE2,bs="cr",k=20) +
                ACCESS + ETHLING, data=comp.sachs)

summary(gdp.lm)

summary(gdp.gam1)
```

GDP Growth, LM Results

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.20e+02   3.76e+01   -3.19  0.00202
OPEN6590     1.86e+00   3.33e-01    5.58  3.0e-07
TROPICS     -7.62e-01   2.70e-01   -2.83  0.00589
LIFE         6.54e+01   1.91e+01    3.42  0.00099
ICRGE80      2.75e-01   7.66e-02    3.59  0.00057
CGB7090      1.22e-01   2.07e-02    5.91  7.7e-08
LGDPEA65    -1.88e+00   1.98e-01   -9.50  7.4e-15
GEAP_POP     9.95e-01   3.30e-01    3.02  0.00340
G7089N       9.97e-02   5.85e-02    1.70  0.09226
SXP         -3.00e+00   9.40e-01   -3.19  0.00203
LIFE2       -7.87e+00   2.45e+00   -3.21  0.00192
ACCESS      -6.03e-01   2.47e-01   -2.44  0.01690
ETHLING     -1.86e-03   3.39e-03   -0.55  0.58382

Residual standard error: 0.797 on 82 degrees of freedom
Multiple R-squared: 0.851,       Adjusted R-squared: 0.829
F-statistic: 38.9 on 12 and 82 DF,  p-value: <2e-16
```

GDP Growth, GAM Results

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 16.83566    1.58590   10.62  5.0e-16
OPEN6590     2.06025    0.31030    6.64  6.5e-09
TROPICS     -0.53903    0.27060   -1.99   0.0504
ICRGE80      0.37233    0.07294    5.10  2.9e-06
LGDPEA65    -2.12289    0.20046  -10.59  5.6e-16
GEAP_POP     0.64165    0.37855    1.70   0.0947
SXP         -2.74615    0.87199   -3.15   0.0024
ACCESS      -0.69192    0.22716   -3.05   0.0033
ETHLING     -0.00329    0.00314   -1.05   0.2977

Approximate significance of smooth terms:
             edf Ref.df     F p-value
s(CGB7090)  3.26   3.26 16.64 1.5e-08
s(G7089N)  11.47  11.47  1.92    0.05
s(LIFE2)    3.89   3.89 12.11 2.3e-07

R-sq.(adj) =  0.881   Deviance explained = 91.5%
GCV score = 0.62295  Scale est. = 0.44187   n = 95
```

How Do We Know Which Is Better?

- Wald tests, worse for GAM: `TROPICS`, `GEAP_POP`; worse for LM: `ETHLING`; the rest have very small p-values for both.
- Scale, GAM: $\sigma^2 = 0.442$, LM: $\sigma^2 = 0.797$.
- Variance explained, GAM: $R^2 = 0.881$, LM: $R^2 = 0.851$.
- Graphing predictors:

```
postscript("Class.Stat.Comp/sachs2.ps")
par(mfrow=c(1,2),mar=c(5,5,1,1),oma=c(1,1,1,1),col.axis="white",col.lab="white",
    col.sub="white",col="black",bg="slategrey",cex.lab=1.5)
plot(comp.sachs$GR6590,gdp.lm$fitted.values,cex=1.05,pch=19,
    xlab="GDP Growth",ylab="LM Fitted Values")
lo.object <- lowess(gdp.lm$fitted.values ~ comp.sachs$GR6590,f=1/5)
lines(lo.object$x,lo.object$y,lwd=2,col="firebrick")
plot(comp.sachs$GR6590,gdp.gam1$fitted.values,cex=1.05,pch=19,
    xlab="GDP Growth",ylab="GAM Fitted Values")
lo.object <- lowess(gdp.gam1$fitted.values ~ comp.sachs$GR6590,f=1/5)
lines(lo.object$x,lo.object$y,lwd=2,col="firebrick")
dev.off()
```