

Generalized Additive Models
oooooooooooo

Multiple Smoothers
oooooooooooooooooooooooooooo

Bivariate Splines
oooooooooooooooooooooooooooo

Module 5: Generalized Additive Models

BIOS 526

Reading

- From Wood, S. *Generalized Additive Models: An Introduction with R, 2nd Edition*: Chapter 4, 5.1.2, 5.3.1, 5.4, 5.5.1, 6.1.2, 6.10, 6.12.
- A nice resource for fitting GAMs:
https://wiki.qcbs.ca/r_workshop8.

Concepts

- Generalized additive models (logistic).
- Additive models with multiple covariates
- Using `gam()` in package `mgcv()` to fit semiparametric models with parametric (slopes) and nonparametric terms (functions)
- Bivariate splines.
- Interpreting non-linear effects.

Generalized Additive Models



Multiple Smoothers



Bivariate Splines



Generalized Additive Models

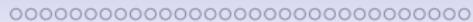
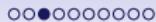
Binary Outcome Example

Recall the example from Module 4:

Dataset: a cohort of live births from Georgia born in the year 2001 (N = 77,340).

Variables:

- ptb : indicator for whether the baby from pregnancy i was born preterm (< 37 weeks).
 - age : the mother's age at delivery.
 - $male$: indicator of the baby's sex (1 = male; 0=female).
 - $tobacco$: indicator for mother's tobacco use during pregnancy (1 = yes; 0 = no)



Previous analysis

```
### Fit logistic regression model  
> fit = glm(ptb~age + male+tobacco, data = dat, family = binomial(link='logit'))  
> summary(fit)
```

Call:

```
glm(formula = ptb ~ age + male + tobacco, family = binomial(link = "logit"),  
    data = dat)
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
-0.5159563	-0.4235975	-0.4102807	-0.4087970	2.2499946	

Coefficients:

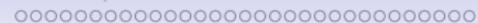
	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-2.421266448	0.063135527	-38.35030	< 2.22e-16 ***
age	-0.000629473	0.002159576	-0.29148	0.7706843
maleM	0.072365862	0.025867177	2.79759	0.0051485 **
tobacco	0.409649486	0.053462733	7.66234	1.8258e-14 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

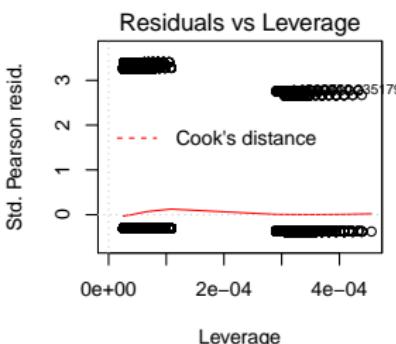
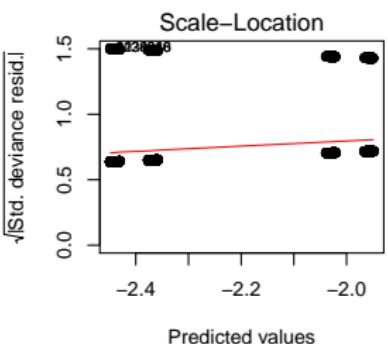
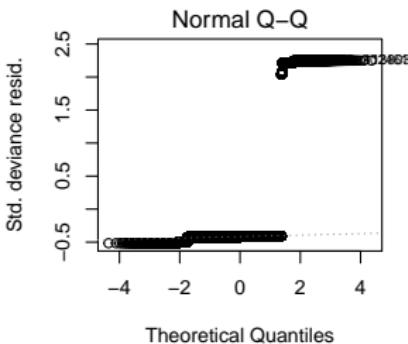
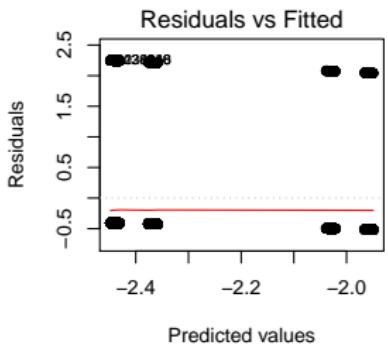
(Dispersion parameter for binomial family taken to be 1)

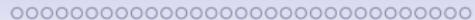
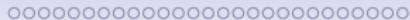
Null deviance: 44907.564 on 77339 degrees of freedom

Residual deviation



The plot diagnostics are not very helpful with binary responses:





Generalized Additive Model

To account for non-linear age effect

$$ptb_i \sim \text{Bernoulli}(p_i)$$

$$\log \left(\frac{p_i}{1 - p_i} \right) = \beta_0 + \beta_1 male_i + \beta_2 tobacco_i + s(age_i),$$

where $s(\text{age}_i)$ is a **smooth** function of age. The above model is known as a **generalized additive model**.

GAMs: Generalized Additive Models.

Everything we learned about additive models is directly applicable in this setting!

GAM with logit link

Using the `mgcv:::gam` function:

```
> fit.gam = gam(ptb~s(age)+male+tobacco,family=binomial,data=dat)
> summary(fit.gam)
```

Family: binomial

Link function: logit

Formula:

```
ptb ~ s(age) + male + tobacco
```

Parametric coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-2.44226	0.01913	-127.647	< 2e-16 ***
maleM	0.07274	0.02588	2.811	0.00494 **
tobacco	0.39016	0.05356	7.284	3.24e-13 ***

Signif. codes: 0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Approximate significance of smooth terms:

edf	Ref.df	Chi.sq	p-value	
s(age)	3.314	4.146	70.17	3.85e-14 ***

Signif. codes: 0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

R-sq.(adj) = 0.00177 Deviance explained = 0.304%

UBRE = -0.42095 Scale est. = 1 n = 77340

GAM diagnostics

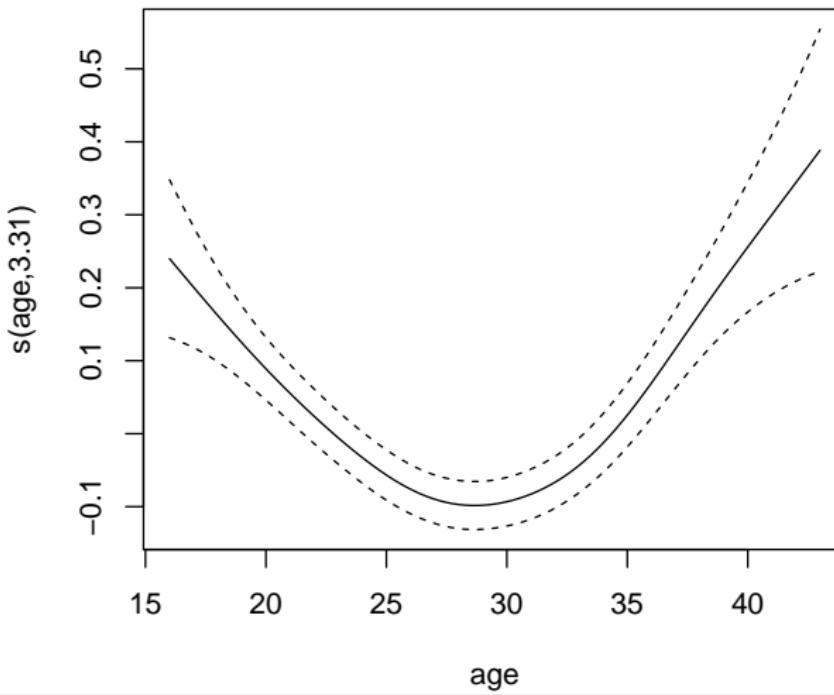
```
> gam.check(fit.gam)
```

```
Method: UBRE    Optimizer: outer newton
full convergence after 3 iterations.
Gradient range [9.816712095e-07,9.816712095e-07]
(score -0.4209494806 & scale 1).
Hessian positive definite, eigenvalue range [1.285936713e-05,1.285936713e-05].
Model rank = 12 / 12
```

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(age)	9.00	3.31	0.94
			0.66

Generalized Additive Model

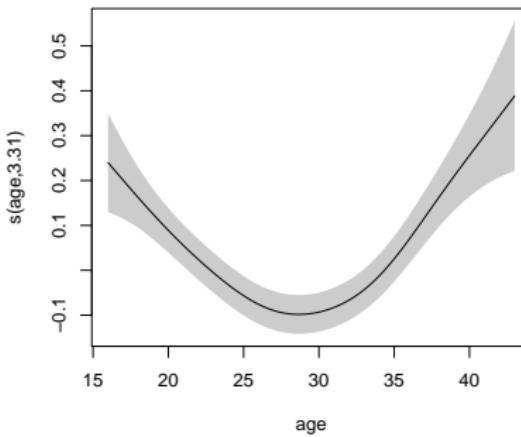


Generalized Additive Model: plots 2

See ?plot.gam for additional options.

Wood suggests coverage probabilities are better with
seWithMean=TRUE to include uncertainty in overall mean, particularly if
edf=1.

```
plot(fit.gam, shade = TRUE, seWithMean = TRUE, pch = 16, cex = 0.5)
```



Bayesian credible intervals

We are assuming the underlying function is smooth. We can formalize this as a prior in a Bayesian model. We won't get into details, but may return to this topic in M7.

In Bayesian statistics, the parameters β are random variables. A ridge penalty corresponds to an improper Gaussian prior:

$$f_{\beta} \propto \exp(-\beta' \mathbf{B} \beta / 2).$$

For Gaussian data, this results in a posterior distribution

$$[\beta | \mathbf{y}, \lambda] \sim N \left\{ \hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X} + \lambda \mathbf{B})^{-1} \right\}$$

For a general likelihood, we use the Fisher Information matrix (Hessian of the negative log likelihood at $\hat{\beta}$)

$$[\beta | \mathbf{y}, \lambda] \sim N \left\{ \hat{\beta}, (\hat{\mathcal{I}} + \lambda \mathbf{B})^{-1} \right\}$$

Multiply by ϕ for quasi-Poisson.

In particular, the "Bayesian credible intervals" plotted in `mgcv::gam` have Frequentist coverage probabilities. See Section 6.10 in Wood.

Multiple Smoothers

Multiple Smooth Terms

Let's consider an additive model for two continuous variables, x_i and z_i :

$$y_i = f_1(x_i) + f_2(z_i) + \epsilon_i, \quad \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2) \quad (1)$$

where $f_1(\cdot)$ and $f_2(\cdot)$ denote smooth relationships between the response y and predictors x and z .

Again, extends to generalized linear models (binomial, Poisson, etc).

We again express non-linear functions using basis functions:

$$f_1(x_i) = \sum_{m=1}^{M_1} \alpha_m b_{m1}(x_i) \quad f_2(z_i) = \sum_{m=1}^{M_2} \beta_m b_{m2}(z_i)$$

Induce smoothing by penalizing regression coefficients.

Note we also use smoothers to control for confounders flexibly.

Associations between Mortality and Fine Particulate Matter

Fine particulate matter ($PM_{2.5}$):

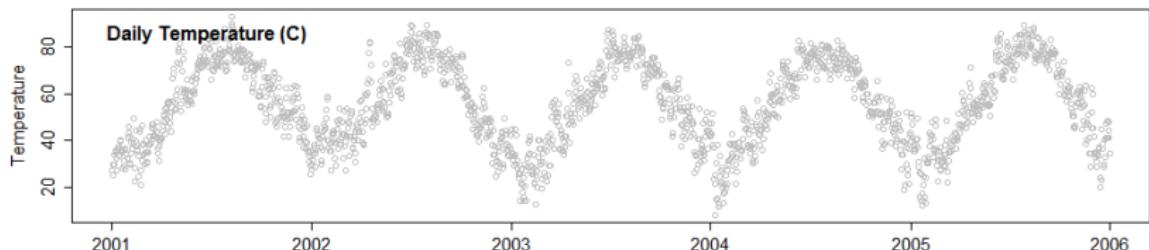
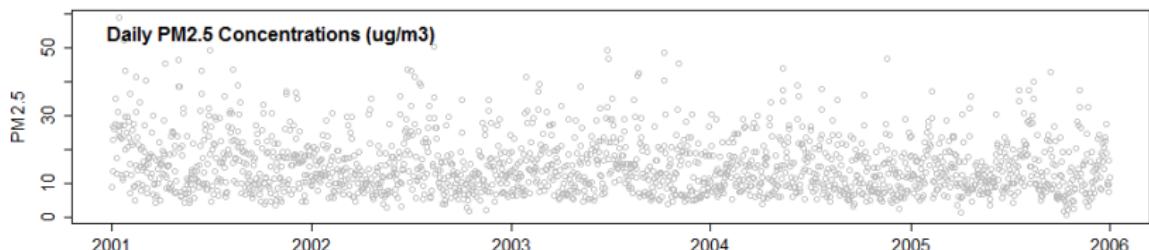
- represents a mixture of solid and liquid particles in the air that are less than $2.5 \mu m$ in diameter;
- mainly arises from combustion sources (power generation, vehicle, and industrial operations).

Scientific Question: what is the association between daily mortality counts and daily concentration of outdoor $PM_{2.5}$ air pollution?

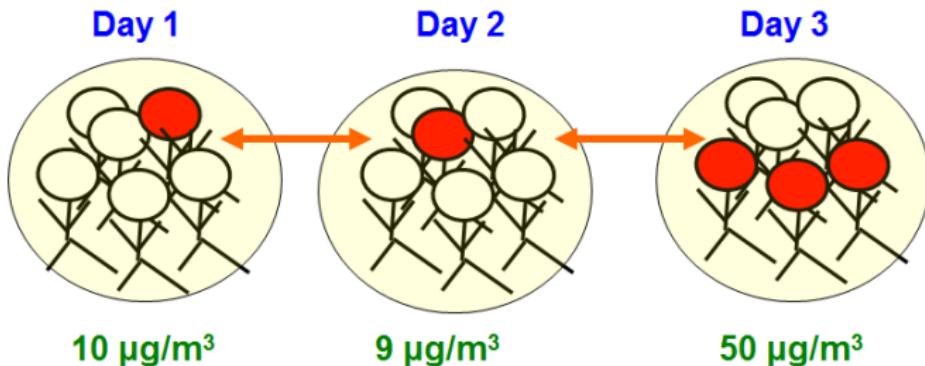
Data Sources:

- Daily counts of non-accidental deaths (age ≥ 65) in the 5-county New York City area (2001-2005) obtained from the National Center for Health Statistics (CDC).
- Daily $PM_{2.5}$ concentrations from Environmental Protection Agency
- Daily meteorology conditions from the National Climatic Data Center (NOAA).

Associations between Mortality and Fine Particulate Matter



Time-Series Health Model



- We are interested in the association between **daily variation** in mortality counts and **daily variation** in exposure.
- In a time-series design we view population as the unit of analysis. I.e. outcome = total mortality counts arising from the population.
- Confounders that vary smoothly in time can be easily controlled for by including long-term time trends.

Time-Series Health Model

Let y_t denote the death count on day t , and x_t be the corresponding PM_{2.5} level. Consider the following model:

$$\log y_t = \beta_0 + \beta_1 x_t + \text{confounders} + \epsilon_i, \quad \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2). \quad (2)$$

Our parameter of interest is β_1 . Here we model log-transformed death counts, which can improve normality of residuals.

Confounding factors to consider:

- Day of the week.
- Seasonality and long-term trends.
- Same-day temperature.
- Same-day dew point temperature.
- Previous day's temperature and dew point temperature.

Data

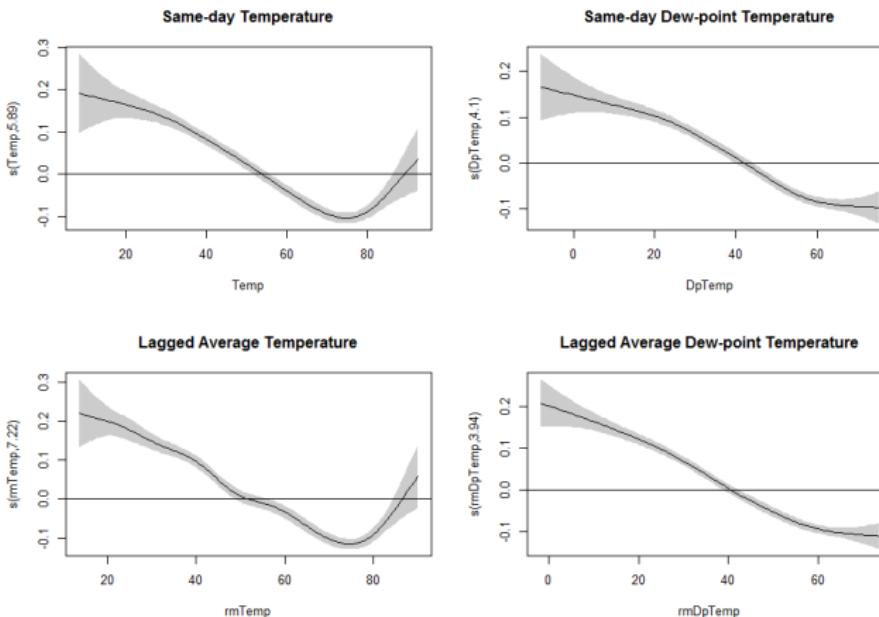
```
> load ("NYC.RData")
> str(health)
'data.frame': 1826 obs. of 12 variables:
 $ date      : Date, format: "2001-01-01" "2001-01-02" "2001-01-03" ...
 $ alldaydeaths : int 171 198 179 169 201 182 167 167 193 159 ...
 $ age65plus   : int 122 146 133 128 145 141 126 116 142 124 ...
 $ cardioresp: int 103 106 109 90 120 101 102 101 115 101 ...
 $ cr65plus    : int 90 92 95 77 98 90 88 82 92 88 ...
 $ dow        : chr "Monday" "Tuesday" "Wednesday" "Thursday" ...
 $ pm25       : num [1:1826, 1] 8.72 13.39 22.9 26.76 25.89 ...
 ...- attr(*, "dimnames")=List of 2
 ... . $ : chr "1" "2" "3" "4" ...
 ... . $ : NULL
 $ Temp       : num 27.6 25.1 25.3 29.8 29.8 33.9 34.9 35.7 32.3 27.6 ...
 $ DpTemp     : num 14.2 11.8 13.1 15.9 20.5 26.7 22.2 31.2 24.4 11.7 ...
 $ rmTemp     : num 27.7 26.8 26 26.7 28.3 ...
 $ rmDpTemp   : num 17.5 14.3 13 13.6 16.5 ...
```

- $DpTemp$ = Dew point temperature.
- rm denotes 3-day running mean of the current day and 2 days prior.
- Note *date* variable is recorded as the in the Date format. We also create a new date variable *date2* which takes value 1 to 1827.

Unadjusted Effects of Meteorology

To educational purposes, examine the unadjusted effect of each potential confounder.

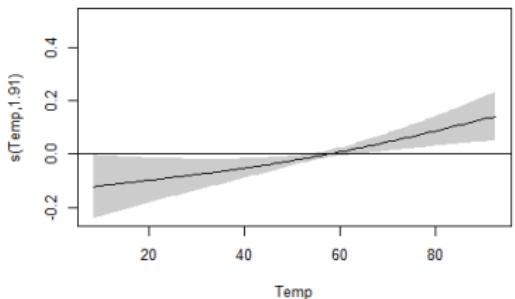
```
> fit1 = gam(log(cr65plus)~s(Temp), data = health)
> plot(fit1, rmse=F, main="Same-day Temperature", shade=T)
```



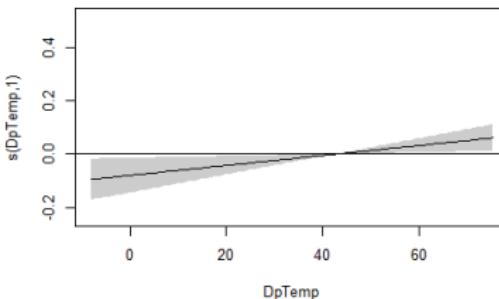
Adjusted Effects of Meteorology

```
> fit = gam(log(alldeaths) ~ s(Temp) + s(DpTemp) + s(rmTemp) + s(rmDpTemp), data = health)
```

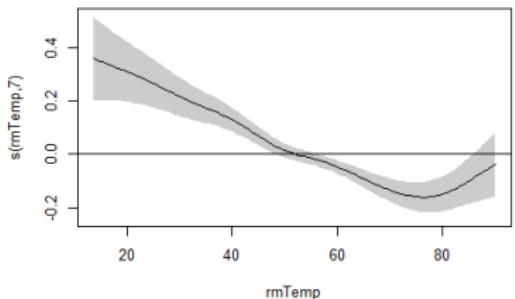
Same-day Temperature



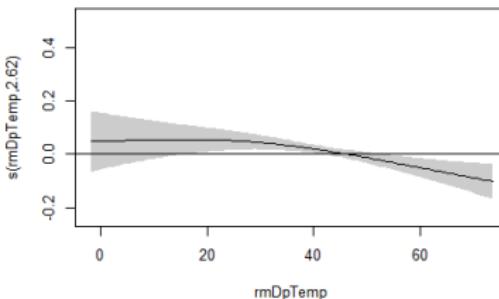
Same-day Dew-point Temperature



Lagged Average Temperature



Lagged Average Dew-point Temperature



Adjusted Effects of Meteorology

```
> summary(fit)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.219108	0.003283	1285	<2e-16 ***

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(Temp)	1.913	2.509	3.797	0.01485 *
s(DpTemp)	1.000	1.000	6.052	0.01398 *
s(rmTemp)	7.003	8.078	7.826	1.84e-10 ***
s(rmDpTemp)	2.622	3.428	3.707	0.00807 **

R-sq.(adj) = 0.32 Deviance explained = 32.5%

GCV score = 0.019829 Scale est. = 0.019682 n = 1826

Note the meteorology variables are highly correlated

```
> round( cor (health[c("Temp", "DpTemp", "rmTemp", "rmDpTemp")]), 2)
      Temp DpTemp rmTemp rmDpTemp
Temp    1.00   0.94   0.97   0.93
DpTemp  0.94   1.00   0.91   0.94
rmTemp  0.97   0.91   1.00   0.96
rmDpTemp 0.93   0.94   0.96   1.00
```

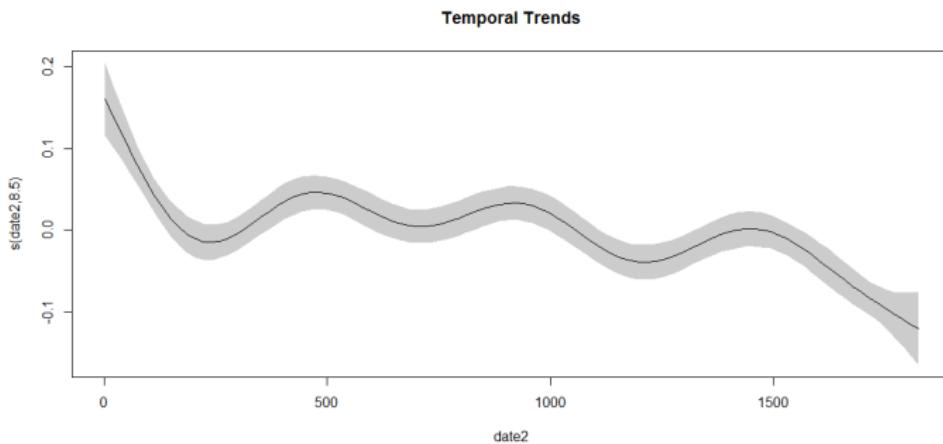
Temporal Trends in Daily Mortality

```
> fit = gam(log(alldeaths)~s(date2), data = health)
> summary(fit)
Parametric coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.219108    0.003839    1099   <2e-16 ***
```

Approximate significance of smooth terms:

edf Ref.df F p-value
s(date2) 8.502 8.93 15.97 <2e-16 ***

R-sq.(adj) = 0.0704 Deviance explained = 7.47%
GCV score = 0.027059 Scale est. = 0.026918 n = 1826



Checking if k is sufficiently large

k -index less than 1 indicates lack of fit.

Similarly, low p-value indicates lack of fit.

```
gam.check(fit)
```

```
Method: GCV  Optimizer: magic
Smoothing parameter selection converged after 9 iterations.
The RMS GCV score gradient at convergence was 5.958159e-07 .
The Hessian was positive definite.
Model rank = 10 / 10
```

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(date2)	9.0	8.5	0.57	<2e-16 ***

Maximum Dimension (*option s(, k =)*)

By default `gam()` assumes the maximum effective degrees of freedom for each smooth effect to be 9. Increase k .

```
> fit = gam(log(alldeaths)~s(date2, k = 20), data = health)

> summary (fit)
Approximate significance of smooth terms:
      edf Ref.df      F p-value
s(date2) 17.65 18.76 55.01 <2e-16 ***

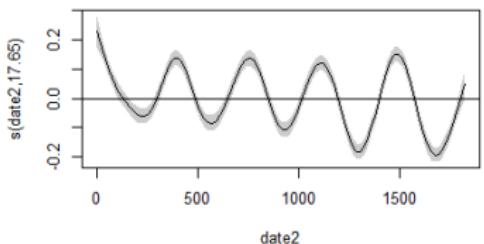
R-sq.(adj) =  0.364  Deviance explained =  37%
GCV score = 0.018602  Scale est. = 0.018412 n = 1826
```

- We increase the adjusted R² significantly.
- The estimated effective DF (edf = 17.54) is still quite close to the upper bound of 20-1. One df is usually lost because of identifiability constraint.

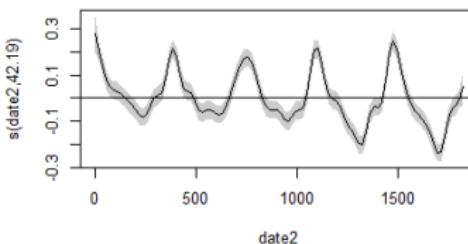
Sensitivity with Maximum Dimension

Because the choice of k is arbitrary, we often need to try different values of k if the estimated edf is close to $k - 1$.

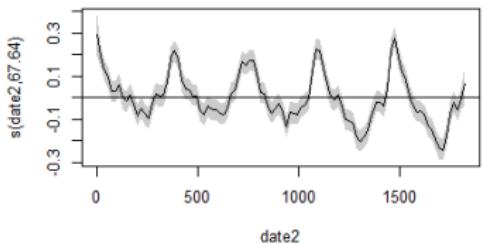
$k = 20; \text{edf} = 17.66$



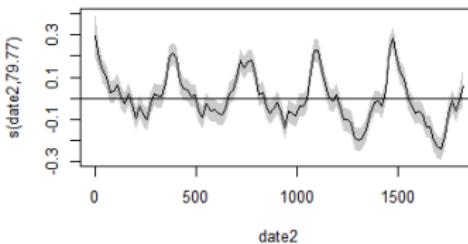
$k = 50; \text{edf} = 42.19$



$k = 100; \text{edf} = 67.64$



$k = 200; \text{edf} = 79.77$



Choosing k

Using `gam.check()`,

```
> gam.check(fit3)
...
      k'  edf k-index p-value
s(date2) 99.0 67.6    0.97   0.095 .
---
> gam.check(fit4)
...
      k'  edf k-index p-value
s(date2) 199.0 79.8    0.99   0.28
> fit5 = gam(log(cr65plus)^s(date2, k = 400), data=health)
> gam.check(fit5)
...
      k'  edf k-index p-value
s(date2) 399.0 83.4    0.99   0.33
```

At $k = 100$ (fit3), $p\text{-value} > 0.05$, but less than 0.10. EDF for $k = 200$ (fit4) similar to $k = 400$ (fit5).

General recommendation: err on the side of a smaller k . (Interpretability and guard against overfitting.)

Un-penalized Splines option $s()$, $fx =$)

`s(, fx = TRUE)` fits **without penalization**.

The results are similar to $k = 200$ with penalization.

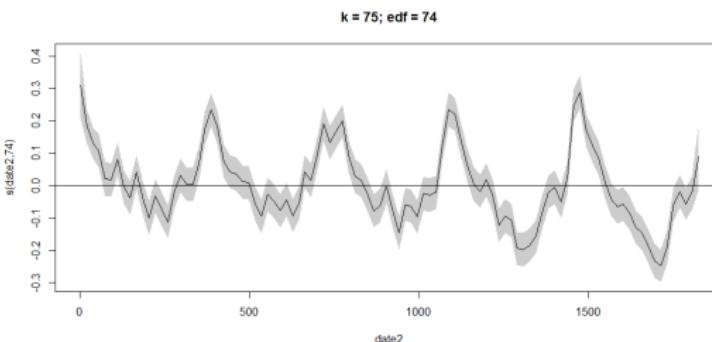
```
> fit = gam(log(alldeaths)~s(date2, k = 75, fx=TRUE), data = health)
> summary(fit)
```

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(date2)	74	74	19.58	<2e-16 ***

R-sq.(adj) = 0.43 Deviance explained = 45.3%

GCV score = 0.017223 Scale est. = 0.016515 n = 1826



Semiparametric Model

Now we examine the linear effect of $PM_{2.5}$ (x_{i1}) on log mortality counts while controlling for DOW_i along with date (x_{i2}), temp (x_{i3}), dew-point temp (x_{i4}), running-mean temp (x_{i5}), and running-mean dew point (x_{i6}):

$$\begin{aligned}y_i = & \beta_0 + \beta_1 x_{i1} + \beta_2 1_{DOW_i=M} + \beta_3 1_{DOW_i=Tu} \cdots + \beta_7 1_{DOW_i=Sa} \\& + f_1(x_{i2}) + f_2(x_{i3}) + f_3(x_{i4}) + f_4(x_{i5}) + f_5(x_{i6}) + \epsilon_i, \\& \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2),\end{aligned}$$

This is called a semiparametric model because it has a mixture of parameteric terms ($PM_{2.5}$ and day of the week) and nonparametric terms.

One reason we may model $PM_{2.5}$ as a linear effect is because our scientific collaborators prefer it.

Semiparametric Model

```
> fit = gam(log(cr65plus)~pm25+factor(dow)+s(date2, k = 100)+s(Temp)+  
           s(DpTemp)+s(rmTemp)+s(rmDpTemp), data = health)  
> summary(fit)  
Parametric coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) 4.2123711446 0.0106328283 396.16657 < 2e-16 ***  
pm25        -0.0003667775 0.0004629403 -0.79228 0.428306  
fdowMonday   0.0336815611 0.0109374584  3.07947 0.002106 **  
fdowSaturday 0.0034936498 0.0109489842  0.31908 0.749701  
fdowSunday   0.0111592226 0.0109906188  1.01534 0.310084  
fdowThursday 0.0147194486 0.0109224826  1.34763 0.177953  
fdowTuesday  0.0107430809 0.0109409288  0.98192 0.326277  
fdowWednesday 0.0121661382 0.0109436453  1.11171 0.266417  
---  
Signif. codes: 0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
```

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(date2)	61.221902	73.813250	6.83228	< 2.22e-16 ***
s(Temp)	1.035084	1.065682	13.72623	0.00015877 ***
s(DpTemp)	1.489371	1.853574	3.41795	0.02627890 *
s(rmTemp)	5.550906	6.775753	3.47986	0.00126217 **
s(rmDpTemp)	1.000003	1.000005	12.80124	0.00035572 ***

After controlling for meteorology, time trends, and day-of-week, there was no statistical significant association between PM_{2.5} and mortality.

Why did we model PM2.5 as linear?

For the purposes of these slides, we modeled PM2.5 as linear. In general, even if your main effect is not a smooth, you may still want to use smooth terms for confounders.

In practice, we should model PM2.5 with a smooth.

Parametric coefficients:

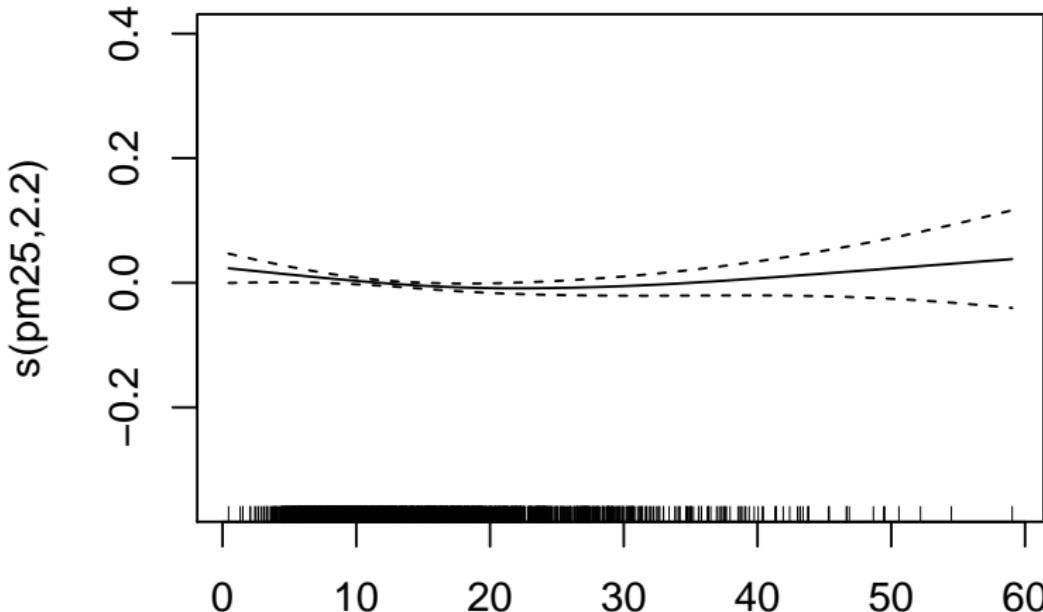
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.207096782	0.007723730	544.69755	< 2.22e-16 ***
fdowMonday	0.033796523	0.010918320	3.09540	0.0019967 **
fdowSaturday	0.003222867	0.010930205	0.29486	0.7681368
...				

Signif. codes: 0 ***? 0.001 **? 0.01 *? 0.05 ?. 0.1 ? ? 1

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(pm25)	2.198333	2.795871	2.52742	0.06878491 .
s(date2)	60.957576	73.527967	6.87685	< 2.22e-16 ***
s(Temp)	1.000013	1.000025	17.43434	3.1367e-05 ***
s(DpTemp)	1.523693	1.908105	3.27180	0.03023256 *
s(rmTemp)	5.537556	6.761898	3.44327	0.00141021 **
s(rmDpTemp)	1.000005	1.000009	13.23867	0.00028215 ***

Smooth term for PM2.5



Lagged PM_{2.5} Exposure

There is typically a *temporal delay* between exposure and outcome.

Let's examine the association between daily mortality and **previous-day** exposure.

```
> health$pm25.lag1 = c(NA, health$pm25[1:1825])
> fit = gam(log(alldeaths)~pm25.lag1+factor(dow)+s(date2, k = 75)+s(Temp)+s(DpTemp)+s(rmTemp)+s(rmDpTemp), data = health)
```

```
> summary (fit)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.2029440	0.0101460	414.248	<2e-16 ***
pm25.lag1	0.0010543	0.0004491	2.347	0.0190 *
factor(dow)Friday	-0.0131336	0.0109570	-1.199	0.2308
factor(dow)Monday	0.0230427	0.0109435	2.106	0.0354 *
/ some outputs deleted			/	

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(date2)	60.138	72.670	6.206	< 2e-16 ***
s(Temp)	1.000	1.000	14.332	0.000158 ***
s(DpTemp)	1.447	1.789	2.037	0.136133
s(rmTemp)	5.827	7.052	2.884	0.005225 **
s(rmDpTemp)	1.000	1.000	9.454	0.002140 **

R-sq.(adj) = 0.463 Deviance explained = 48.6%

GCV score = 0.016212 Scale est. = 0.015524 n = 1825

Lagged PM_{2.5} Exposure smooth term

```
> fit = gam(log(cr65plus)~s(pm25.lag1)+fdow+s(date2, k = 100 )+s(Temp)+ s(DpTemp)+s(rmTemp)
> summary(fit)

Family: gaussian
Link function: identity

Formula:
log(cr65plus) ~ s(pm25.lag1) + fdow + s(date2, k = 100) + s(Temp) +
    s(DpTemp) + s(rmTemp) + s(rmDpTemp)

Parametric coefficients:
              Estimate Std. Error   t value Pr(>|t|)    
(Intercept) 4.205748765 0.007731497 543.97601 < 2.22e-16 ***
fdowMonday  0.036176289 0.010980266   3.29466  0.0010051 **  
fdowSaturday 0.004256722 0.010918397   0.38987  0.6966825    
...
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Approximate significance of smooth terms:
          edf  Ref.df     F  p-value    
s(pm25.lag1) 1.000040 1.000080 5.50991 0.01902030 *  
s(date2)      60.137293 72.669727 6.29239 < 2.22e-16 *** 
s(Temp)       1.000002 1.000004 14.33175 0.00015864 *** 
s(DpTemp)     1.446148 1.787928 2.05017 0.09521592 .  
s(rmTemp)     5.826145 7.051150 2.86630 0.00532417 ** 
s(rmDpTemp)   1.000004 1.000007 9.45312 0.00214032 ** 
```

Note on time series data

We should analyze the residuals for evidence of autocorrelation.

If residuals are correlated, e.g., nearby residuals are more similar than distant residuals, our inference is not valid, and p-values tend to be too small.

This is beyond the scope of this course, but a nice tutorial is available at
<https://petolau.github.io/>

[Analyzing double seasonal time series with GAM in R](#)

Coefficient Interpretation

- Daily counts of non-accidental mortality was positively associated with previous-day PM_{2.5} level among those 65 or above.
- We estimated a 0.00105 (standard error = 0.00045) increase in log death count per unit ($\mu\text{g}/\text{m}^3$) increase in PM_{2.5} level in the previous day.
- Note that when modeling the outcome on a log-scale

$$\log y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

then β_1 is the change in $E[\log y_i]$ per unit change in x_i . Therefore, a 10 unit increase in PM_{2.5} level was associated with an increase in daily death counts by $e^{(0.00105 \times 10)} = 1.0105$, with 95% confidence interval (1.0017, 1.0195).

Sensitivity Analysis

In the above analysis, we selected smoothness parameters for confounders automatically using GCV. However,

- The GCV criterion used is aimed to maximize prediction performance.
- In a health analysis, our goal is not to construct a model that **best predicts** the outcome, but to obtain accurate **health effect association** where confounding is minimized.

Sensitivity Analysis Approach

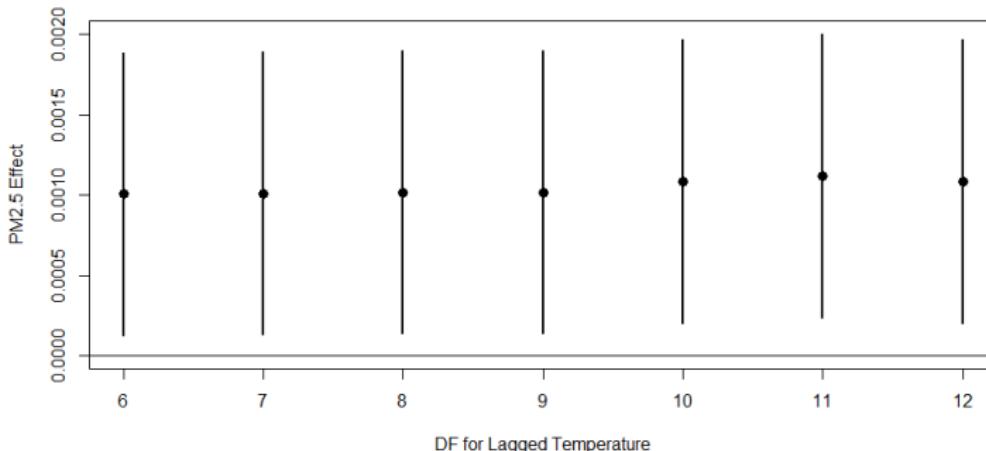
- Starting with the estimated edf for each confounder, we will increase each of them and refit the model.
- We will examine how the PM_{2.5} estimate changes as we allow more flexible control of confounders.
- We hope to see the effect estimates to be **stable and robust** against different ways to specific confounder effects.

Sensitivity Analysis: Lagged Temperature Effect

```
> Results = NULL
> for (df in 6:12){
  fit = gam(log(alldeaths)~pm25.lag1+factor(dow)+s(date2, k = 100 )+s(Temp)+s(DpTemp)+s(rmTemp, k = df, fx=TRUE )+s(rmDpTemp), data = health)

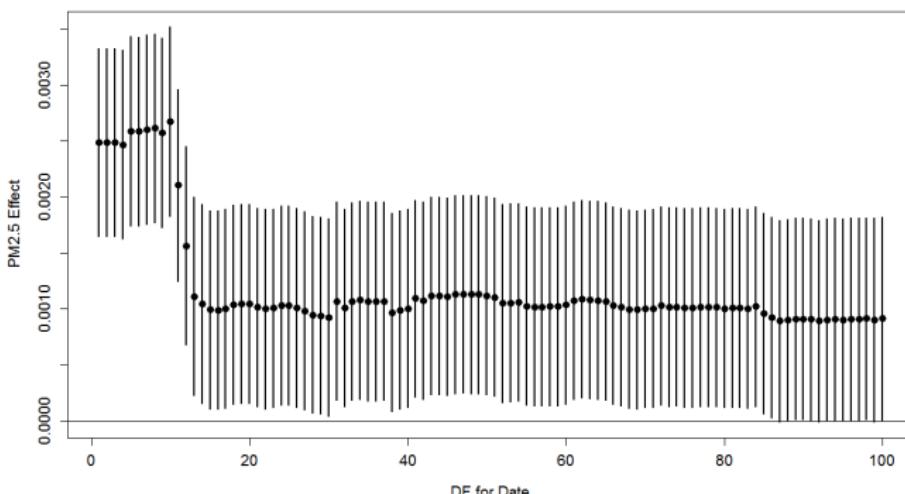
  #Extract PM2.5 point estimate and standard error
  est = c(summary(fit)$p.coef[2], (fit)[2])

  #Save results
  Results = rbind (Results, est)
}
}
```



Sensitivity Analysis: Temporal Trends

- One challenge in time-series analysis is how much control do we need for temporal trend.
- Temporal trend is used to account for **unmeasured confounders** that vary in time, often at different temporal scales.
- Note below how the estimates *stabilize* after there is sufficient control for temporal trends. Too little temporal control will result in an over-estimate of the PM_{2.5}-mortality association.



Generalized Additive Models
oooooooooooo

Multiple Smoothers
oooooooooooooooooooooooooooo

Bivariate Splines
●oooooooooooooooooooo

Bivariate Splines

Bivariate Splines

Under an additive model framework, we can also consider smooth effects of two variables jointly:

$$y_i = f(x_i, z_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2).$$

We can think of $f(x_i, z_i)$ as a **surface**.

We can use 2-dimensional splines to model $f(x_i, z_i)$.

Let $\mathbf{s}_i = (x_i, z_i)$ be some pair of covariate values, and let $\mathbf{k}_m = (x_m, z_m)$ denote the m^{th} knot in the domain of x_i and z_i . We can express the smooth function as

$$f(x_i, z_i) = \beta_0 + \sum_{m=1}^M \beta_m b_m(\mathbf{s}_i; \mathbf{k}_m).$$

Note that $b_m(\cdot, \cdot)$ is a basis function that maps $\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.

Thin-plate Spline

One popular bivariate basis function uses **thin-plate splines**, which extends to $\mathbf{s}_i \in \mathbb{R}^d$ and $\partial^l g$ penalties. We consider $d = 2$ and $l = 2$:

$$f(\mathbf{s}_i) = \beta_0 + \beta_1 x_t + \beta_2 z_t + \sum_{m=1}^M \beta_{2+m} b_m(\mathbf{s}_i; \mathbf{k}_m)$$

using the radial basis:

$$b_m(\mathbf{s}_i; \mathbf{k}_m) = \|\mathbf{s}_i - \mathbf{k}_m\|^2 \log(\|\mathbf{s}_i - \mathbf{k}_m\|).$$

Here, $\|\mathbf{s}_i - \mathbf{k}_m\|$ is the Euclidean distance between the covariate \mathbf{s}_i and the knot location \mathbf{k}_m .

The radial basis kernel is $r^2 \log r$.

The thin-plate spline is sensitive to the scale of each variable, but invariant to rotation (isotropic).

It is best for variables measured on the same scale (e.g. geographical distance).

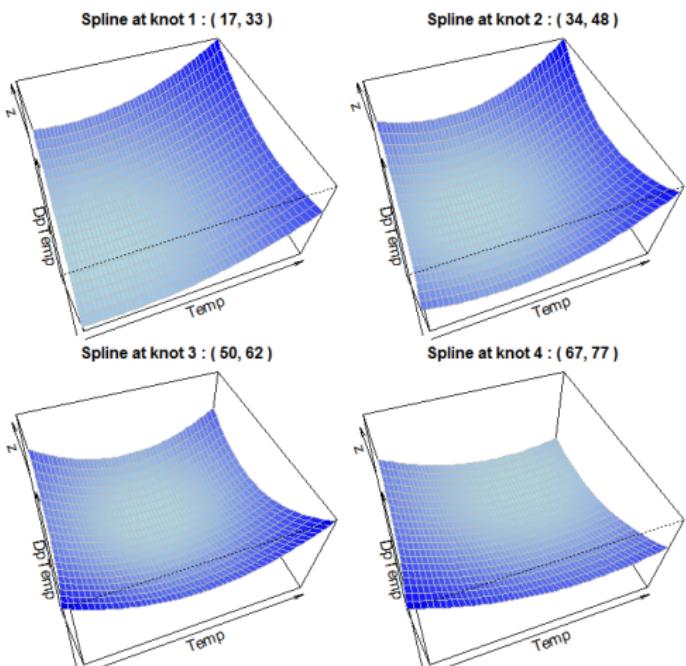
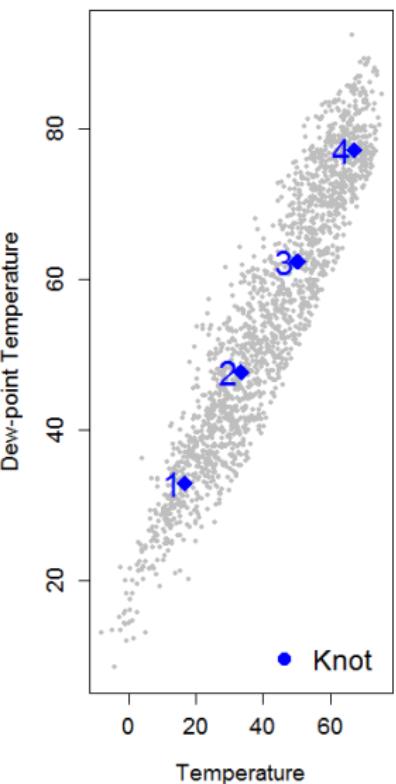
Thin-plate Spline, cont.

It can be shown that the thin-plate spline function minimizes

$$\sum_{i=1}^n \{y_i - f(x_i, z_i)\}^2 + \lambda \int \left(\frac{\partial^2 f(x, z)}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f(x, z)}{\partial x \partial z} \right)^2 + \left(\frac{\partial^2 f(x, z)}{\partial z^2} \right)^2 dx dz .$$

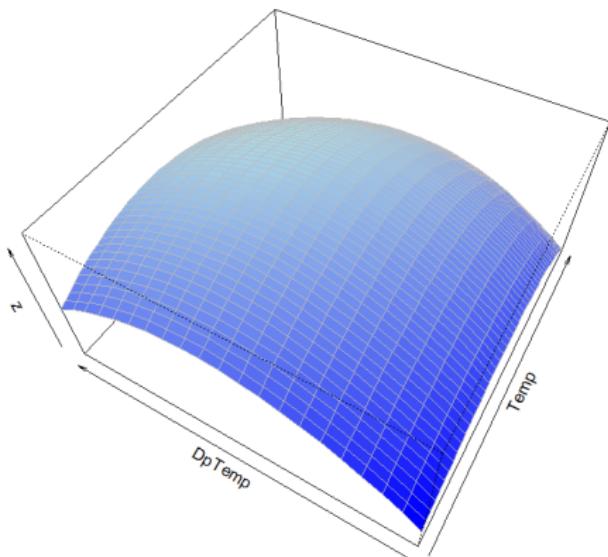
More information is in Wood 2017, pages 215-221, and references therein.

Thin-plate Spline

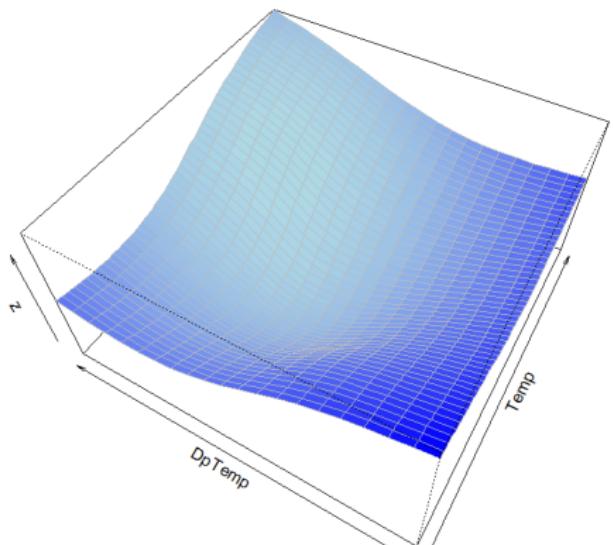


Thin-plate Spline

Basis coefficients = [-2,1,1,-2]



Basis coefficients = [1,-1,-1,1]



Thin-plate regression spline

`mgcv::gam` uses thin-plate **regression** splines as the default for smoothers of a single variable (as well as two variables).

This is implemented in a ‘knot-free’ manner.

This is the general idea.

For $d = 2, l = 2$:

1. Construct the $n \times n$ matrix \mathbf{E} from $\|\mathbf{s}_i - \mathbf{s}_{i'}\|^2 \log(\|\mathbf{s}_i - \mathbf{s}_{i'}\|)$.
2. Use the singular value decomposition to find a low rank representation, e.g., k leading singular vectors, and use this in place of \mathbf{X} in the penalized objective function.
3. In practice, there are some additional things to worry about to make $\mathbf{1}$ (for the intercept), \mathbf{x} and \mathbf{z} in the null space of the penalty.
4. Then estimate the β_0, β_1 for \mathbf{x} , β_2 for \mathbf{z} (unpenalized) and β_3, \dots, β_k (penalized), which dramatically reduces computation costs.

The formulas for the general case (d dimensions and l th derivative) get a bit complicated; see 5.5 in Wood.

Joint Effects of Temperature and Dew point Temperature

To define a bivariate smoother, simply specify $s(var_1, var_2)$ in the equation formula.

Default in `mgcv:gam` is to use a thin-plate regression spline.

Let's first look at the joint effects of same-day temperature and dew point temperature on log mortality, controlling only for time trends.

```
> fit1 = gam(log(alldeaths)~s(date2, k=100) + s(Temp, DpTemp, k = 20), data = health)
> summary (fit1)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.219108	0.002953	1429	<2e-16 ***

Approximate significance of smooth terms:

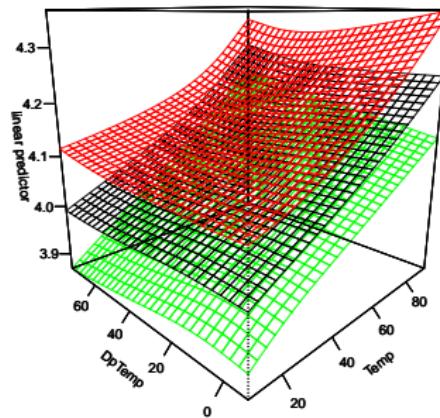
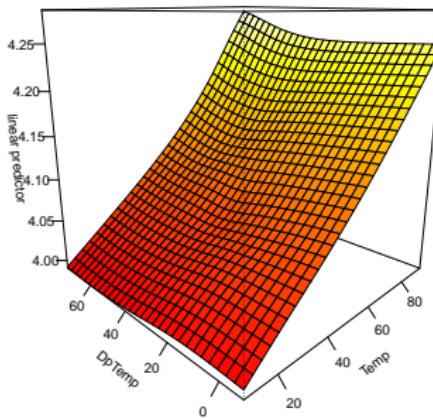
	edf	Ref.df	F	p-value
s(date2)	69.527	82.144	8.501	< 2e-16 ***
s(Temp,DpTemp)	5.938	8.032	6.407	2.96e-08 ***

R-sq.(adj) = 0.45 Deviance explained = 47.3%
 GCV score = 0.016624 Scale est. = 0.015928 n = 1826

You should also use `gam.check` (note shown here)

Joint Effects of Temperature and Dew Point Temperature

Perspective Plots from Thin-Plate Spline



red/green are +/- 2 s.e.

Tensor Product

Another way to obtain a 2-D spline is by construction. First consider the effect of a variable x_i specified by M basis functions $b_m(x_i)$

$$f(x_i) = \sum_{m=0}^M \beta_m b_{m,x}(x_i).$$

Now assume $f(x_i, z_i)$ is created by allowing each spline coefficient to vary with the second variable z_i : M

$$f(x_i, z_i) = \sum_{m=0}^M \beta_m(z_i) b_{m,x}(x_i).$$

We can express $\beta_m(z_i)$ also as a smooth function of z_i using N basis functions $b_{n,z}(z_i)$:

$$f(x_i, z_i) = \sum_{m=0}^M \sum_{n=0}^N \alpha_{m,n} b_{n,z}(z_i) b_{m,x}(x_i).$$

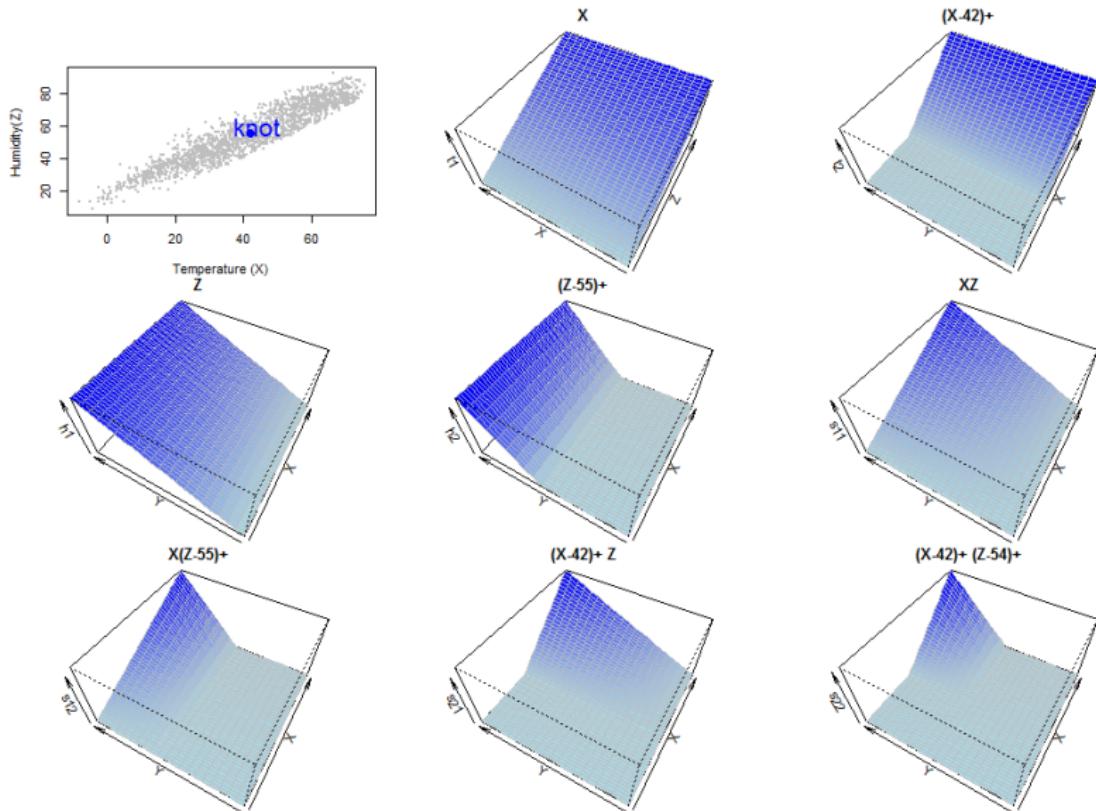
This is equivalent to expressing $f(x_i, z_i)$ as all pairwise basis functions of x_i and z_i :

$$f(x_i, z_i) = \sum_{m=0}^M \sum_{n=0}^N \alpha_{m,n} b_{n,z}(z_i) b_{m,x}(x_i).$$

Tensor Product

Example with 1 knot: write out all pairwise interactions

Tensor Product Example



Tensor Product

The roughness penalty optimized by the tensor product is

$$\sum_{i=1}^n \{y_i - f(x_i, z_i)\}^2 + \int \lambda_x \left(\frac{\partial^2 f(x, z)}{\partial x^2} \right)^2 + \lambda_z \left(\frac{\partial^2 f(x, z)}{\partial z^2} \right)^2 dx dz .$$

- Allows penalization in each variable dimension by assigning a smoothing parameter for each marginal smooth effect.
- Choice of basis function and knots also do not need to be the same for each covariate.
- Provides a recipe for constructing flexible multivariate joint effects.
- Often used for modeling interactions where the degree of smoothness may not be the same for all covariates.

Fitting Tensor Product

A bivariate smoother using tensor product is specified by `te(var_1, var_2, bs='cr')` in the equation formula.

Recommended when scales differ.

The option `bs = 'cr'` specifies using cubic spline functions for each covariate .

```
> fit1 = gam (log(alldeaths) ~ s(date2, k = 100) +
   te(Temp, DpTemp, k = 20, bs = "cr") , data = health)

> summary (fit1)

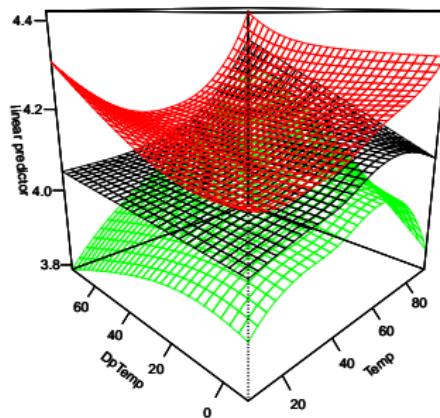
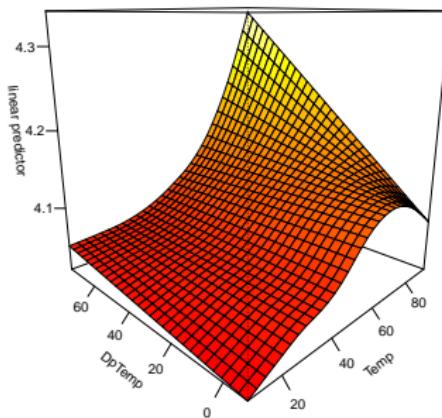
Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.219108   0.002952   1429   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
            edf Ref.df      F p-value    
s(date2)      68.353 81.039  8.343 <2e-16 ***
te(Temp,DpTemp) 6.003  7.391 18.134 <2e-16 ***

R-sq.(adj) =  0.45  Deviance explained = 47.3%
GCV score = 0.016601  Scale est. = 0.015916 n = 1826
```

Joint Effects of Temperature and Dew Point Temperature

Perspective Plots from Cubic Tensor Product Spline

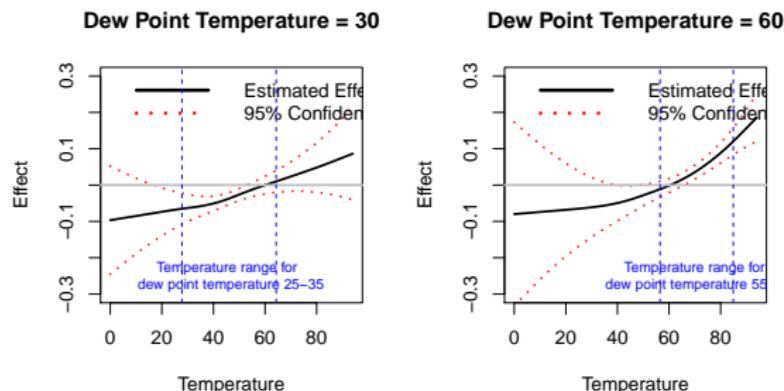


red/green are +/- 2 s.e.

Marginal Effects

By fitting an effect surface, we can obtain **slices** with respect to a covariate and examine the **conditional** effect of another covariate.

For example, we can look at the non-linear effect of temperature when dew point temperature = 30 versus 60.



PM_{2.5} and Temperature

An important research question is whether there is an interaction between air pollution and temperature.

Let's extend our previous model to consider previous-day PM_{2.5} level and 3-day moving average of temperature. Note here we use thin-plate splines, which are much faster to fit than tensor; pm25.lags and rmTemp are very roughly on same scale.

```
> fit = gam(log(alldeaths)~s(pm25.lag1, rmTemp)+factor(dow)+s(date2, k = 75)+  
           s(DpTemp)+s(rmDpTemp), data = health)  
> summary (fit)
```

Parametric coefficients:

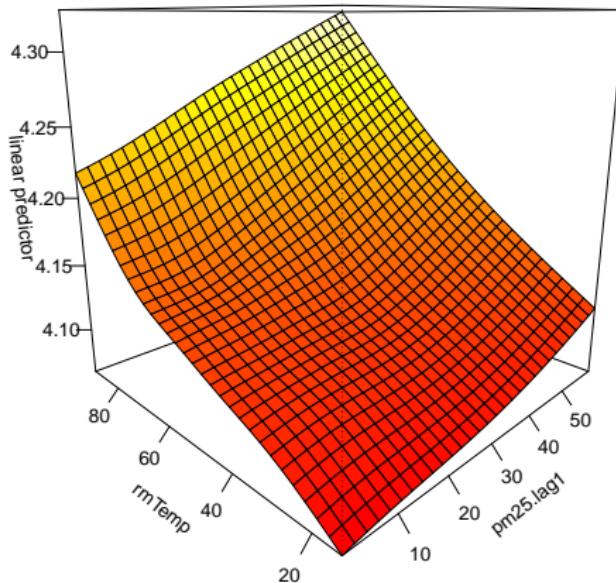
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.2191876	0.0077934	541.380	<2e-16 ***
factor(dow)Friday	-0.0142658	0.0110132	-1.295	0.1954
factor(dow)Monday	0.0222996	0.0110146	2.025	0.0431 *
factor(dow)Saturday	-0.0092121	0.0110082	-0.837	0.4028
factor(dow)Thursday	0.0011288	0.0110248	0.102	0.9185

Approximate significance of smooth terms:

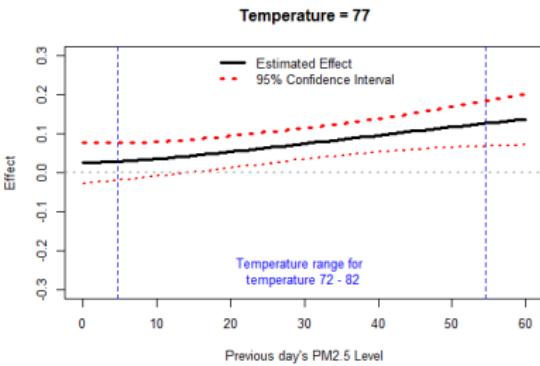
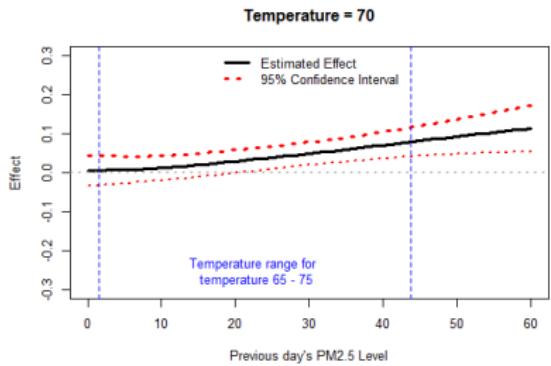
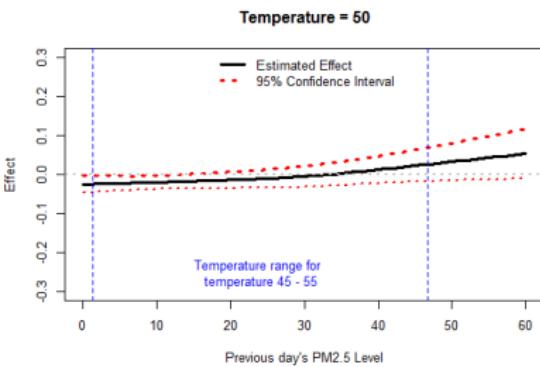
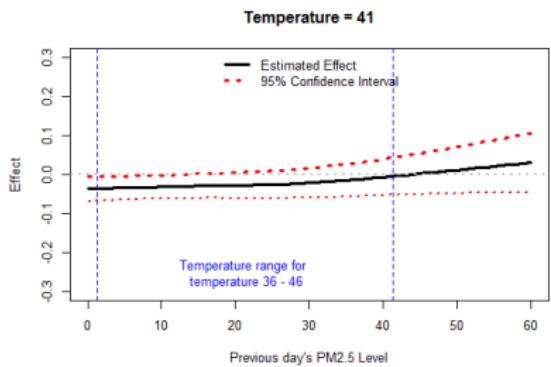
	edf	Ref.df	F	p-value
s(pm25.lag1,rmTemp)	5.858	8.381	4.048	6.35e-05 ***
s(date2)	57.808	66.508	6.721	< 2e-16 ***
s(DpTemp)	1.362	1.633	25.075	8.11e-10 ***
s(rmDpTemp)	1.000	1.000	35.085	3.79e-09 ***

Joint Effects of Temperature and PM_{2.5}

Effect of temperature is more pronounced than PM_{2.5}.



PM_{2.5} Effect Modification by Temperature?



Extracting Effects from Smooth Function

- Let's say we are interested in the difference in $f(x_i)$ when x_i increases from value a to value b :

$$\hat{f}(b) - \hat{f}(a) = \sum_{m=1}^M \hat{\beta}_m b_m(b) - \sum_{m=1}^M \hat{\beta}_m b_m(a) = \sum_{m=1}^M \hat{\beta}_m \{b_m(b) - b_m(a)\}.$$

The covariance matrix is given by

$Cov(\mathbf{B}'_{b-a}\hat{\beta}) = \mathbf{B}'_{b-a}Cov(\hat{\beta})\mathbf{B}_{b-a}$, where

$$\mathbf{B}_{b-a} = [b_1(b) - b_1(a), b_2(b) - b_2(a), \dots, b_M(b) - b_M(a)]'.$$

- However $b_m()$'s are basis functions and $b_m(b) - b_m(a) \neq b_m(b - a)$. We will need to construct the new covariate vector \mathbf{B} ourselves.

Extracting Effects from a Smooth Function

We want to estimate the effect of a $10 \mu\text{g}/\text{m}^3$ increase in PM_{2.5} levels from 40 to $50 \mu\text{g}/\text{m}^3$ when temperature = 70F.

```
> X1 = predict(fit, data.frame(pm25.lag1 = 40, rmTemp=70,
+                               DpTemp=0, rmDpTemp = 0, dow = "Sunday", date2=0),type= "lpmatrix")
> X2 = predict(fit, data.frame(pm25.lag1 = 50, rmTemp=70,
+                               DpTemp=0, rmDpTemp = 0, dow = "Sunday", date2=0),type= "lpmatrix")

> X.diff = X2 - X1
> dim (X.diff)
[1] 1 128
> Est = X.diff %*% coef (fit) ## Estimate
> se = sqrt( X.diff %*% vcov (fit) %*% t(X.diff) ) ## Standard Error
> Est; se
      [,1]
1 0.02213508
      1
1 0.01096592
```

So our estimate is 0.022 (95%CI 0.001, 0.044).

The same effect of a 10-unit increase in PM_{2.5} levels from 20 to $30 \mu\text{g}/\text{m}^3$ is 0.020 (95%CI 0.004, 0.036).

These two estimates are very similar because the PM_{2.5} effect appears to be quite linear.

Inference in models with smoothing splines

For parametric terms, the inference is identical to purely parametric model.

You can use the t-statistics from `summary(fit)`.

```
> fit.full = gam(log(cr65plus)~s(pm25.lag1, rmTemp)+fdow+s(date2, k = 100)+s(DpTemp)+s(rmDpTemp))  
> summary(fit.full)
```

Family: gaussian

Link function: identity

Formula:

```
log(cr65plus) ~ s(pm25.lag1, rmTemp) + fdow + s(date2, k = 100) +  
    s(DpTemp) + s(rmDpTemp)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.219093	0.007784	541.997	<2e-16 ***
fdowFriday	-0.014166	0.011001	-1.288	0.1980
fdowMonday	0.022352	0.011002	2.032	0.0423 *
fdowSaturday	-0.009111	0.010995	-0.829	0.4074
fdowThursday	0.001261	0.011012	0.114	0.9089
fdowTuesday	-0.001689	0.011002	-0.153	0.8780
fdowWednesday	0.000470	0.011012	0.043	0.9660

Signif. codes: 0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Inference in models with smoothing splines

For overall effect of fdow, use anova:

```
> anova(fit.full)
```

Family: gaussian

Link function: identity

Formula:

```
log(cr65plus) ~ s(pm25.lag1, rmTemp) + fdow + s(date2, k = 100) +
  s(DpTemp) + s(rmDpTemp)
```

Parametric Terms:

df	F	p-value
fdow	6	2.154
		0.0448

Inference in models with smoothing splines

For smoothed terms, the inference is approximate because the distribution of the test statistics is impacted by the penalization.

The idea is to jointly test the significance of the $\hat{\beta}_j$ for the coefficients corresponding to the spline of the j th smooth term.

There is some discussion of these approximate p-values in
`?summary.gam`. For detailed discussion, see Wood 2017 p.304.

Approximate inference for smooth effects:

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value	
s(pm25.lag1,rmTemp)	5.433	7.738	4.246	6.30e-05	***
s(date2)	62.047	74.627	6.118	< 2e-16	***
s(DpTemp)	1.210	1.379	29.260	2.89e-09	***
s(rmDpTemp)	1.000	1.000	34.655	4.68e-09	***

R-sq.(adj) = 0.458 Deviance explained = 48%
GCV = 0.016377 Scale est. = 0.015688 n = 1825

Inference in models with smoothing splines

Should we use a bivariate spline for pm25.lag1 and rmTemp, or two univariate splines? Don't do this:

```
> anova(fit.reduced2,fit.full,test='F')
Analysis of Deviance Table

Model 1: log(cr65plus) ~ s(pm25.lag1) + s(rmTemp) + s(date2, k = 100) +
  fdow + s(DpTemp) + s(rmDpTemp)
Model 2: log(cr65plus) ~ s(pm25.lag1, rmTemp) + fdow + s(date2, k = 100) +
  s(DpTemp) + s(rmDpTemp)
  Resid. Df Resid. Dev          Df      Deviance F Pr(>F)
1 1734.9095 27.353616
2 1733.2552 27.428321 1.6542684 -0.074705162
```

What happened here?

Models are not nested. Works better with tensor spline, which we can construct in a special way to make nested. Then we can recast the problem as testing for an interaction.

Inference in models with smoothing splines

Before testing for interaction, let's do an approximate anova for date2.

```
> fit.nodate2 = gam(log(cr65plus)~s(pm25.lag1,rmTemp)+fdow+s(DpTemp)+s(rmDpTemp), data = h
> anova(fit.nodate2,fit.full,test='F')
Analysis of Deviance Table

Model 1: log(cr65plus) ~ s(pm25.lag1, rmTemp) + fdow + s(DpTemp) + s(rmDpTemp)
Model 2: log(cr65plus) ~ s(pm25.lag1, rmTemp) + fdow + s(date2, k = 100) +
  s(DpTemp) + s(rmDpTemp)
  Resid. Df Resid. Dev      Df Deviance      F    Pr(>F)
1     1798.9    34.854
2     1733.3    27.428 65.639    7.4257 7.2111 < 2.2e-16 ***
> ((34.854 - 27.428)/(1798.9-1733.3)) / (27.428 / 1733.3)
[1] 7.1537
```

Testing for an interaction

To test for an interaction between pm25.lag1 and rmTemp, we can construct the tensor spline in a special way such that the univariate splines are in the null space of the penalty of the tensor spline. See pages 243 and the example on page 343-346.

```
fit.full.tensor = gam(log(cr65plus)~s(pm25.lag1,bs='cr')+s(rmTemp,bs='cr')  
+ti(pm25.lag1, rmTemp,bs='cr')+fdow+s(date2, k = 100)+s(DpTemp)+s(rmDpTemp), data = health)  
summary(fit.full.tensor)  
anova(fit.full.tensor)
```

Testing for an interaction

```
> anova(fit.full.tensor)
```

Family: gaussian

Link function: identity

Formula:

```
log(cr65plus) ~ s(pm25.lag1, bs = "cr") + s(rmTemp, bs = "cr") +
  ti(pm25.lag1, rmTemp, bs = "cr") + fdow + s(date2, k = 100) +
  s(DpTemp) + s(rmDpTemp)
```

Parametric Terms:

df	F	p-value
fdow	6	2.24
		0.0371

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(pm25.lag1)	1.000	1.001	4.147	0.0418
s(rmTemp)	5.301	6.480	2.524	0.0171
ti(pm25.lag1,rmTemp)	1.000	1.000	3.463	0.0629
s(date2)	59.917	72.436	6.143	<2e-16
s(DpTemp)	1.000	1.000	41.860	<2e-16
s(rmDpTemp)	1.000	1.000	31.651	<2e-16

H_0 : $g_{xz}(x_i, z_i)$ does not improve model fit over $g_x(x_i)$ and $g_z(z_i)$.
 $p=0.06$.

Inference in models with smoothing splines

Let's refit the model with pm25.lag1 as a linear term.

```
> fit.reduced3 = gam(log(cr65plus)~pm25.lag1+s(rmTemp)
+ s(date2, k = 100)+fdow+s(DpTemp)+s(rmDpTemp), data = health)
> summary(fit.reduced3)

Family: gaussian
Link function: identity

Formula:
log(cr65plus) ~ pm25.lag1 + s(rmTemp) + s(date2, k = 100) + fdow +
  s(DpTemp) + s(rmDpTemp)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.203e+00	1.018e-02	412.689	<2e-16 ***
pm25.lag1	1.073e-03	4.508e-04	2.381	0.0174 *
fdowFriday	-1.478e-02	1.099e-02	-1.345	0.1789
fdowMonday	2.249e-02	1.098e-02	2.048	0.0407 *
fdowSaturday	-9.026e-03	1.098e-02	-0.822	0.4110
fdowThursday	5.425e-04	1.100e-02	0.049	0.9607
fdowTuesday	-1.433e-03	1.098e-02	-0.130	0.8962
fdowWednesday	9.336e-05	1.100e-02	0.008	0.9932

...
R-sq.(adj) = 0.459 Deviance explained = 48.2%
GCV = 0.016324 Scale est. = 0.015642 n = 1825

GAM or linear?

It may seem like we can test for whether or not to include a smooth term versus linear term using anova. However, the anova test can produce funny results when the EDF is approximately one, as the change in DF is very small. So, I suggest not doing this:

```
> anova(fit.reduced2,fit.reduced3,test='F')
Analysis of Deviance Table

Model 1: log(cr65plus) ~ s(pm25.lag1) + s(rmTemp) + s(date2, k = 100) +
  fdow + s(DpTemp) + s(rmDpTemp)
Model 2: log(cr65plus) ~ pm25.lag1 + s(rmTemp) + s(date2, k = 100) + fdow +
  s(DpTemp) + s(rmDpTemp)
Resid. Df Resid. Dev      Df   Deviance      F Pr(>F)
1     1734.9    27.354
2     1734.9    27.353 0.016157 0.00029882 1.1824 0.0325 *
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
```

GAM or linear?

When you have an effect that is close to linear, your interpretation is very similar whether or not you use a gam, and when EDF=1, then the approximate p values are equivalent to refitting with a linear term.

To determine whether or not to include a smooth or linear effect, I suggest looking at the EDF. If it is greater than 1, than it seems reasonable to use a smooth term.

There is not really any disadvantage to modeling with a smooth term, except for some extra work we need to do for interpretation.

One approach to test for a non-linear effect is to construct a special spline that separates the linear and non-linear parts:

[https://stats.stackexchange.com/questions/449641/
is-there-a-hypothesis-test-that-tells-us-whether-we-should-use-](https://stats.stackexchange.com/questions/449641/is-there-a-hypothesis-test-that-tells-us-whether-we-should-use-)