Methods of Applied Stats, Non-parametric and semi-parametric models

Patrick Brown, University of Toronto

Sept to Dec 2020

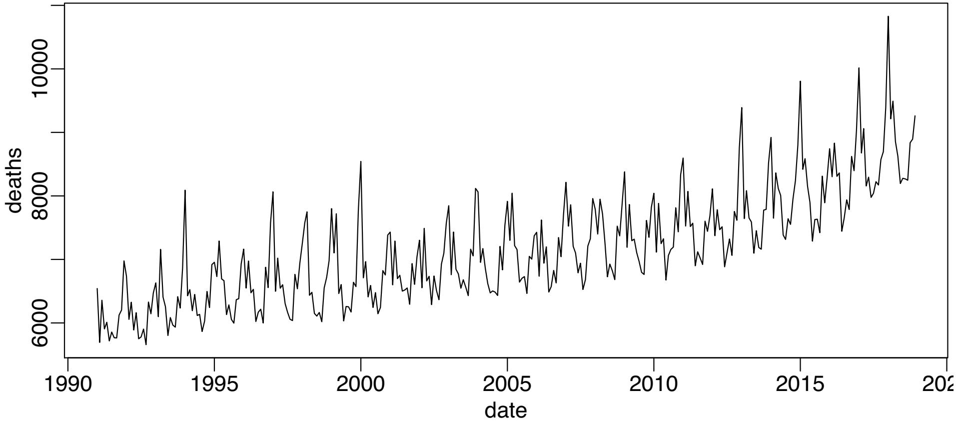
Non-parametric modelling

also known as fitting wiggly lines through points

- Penalized least squares
- Bayesian semi-parametrics with INLA

Deaths in Ontario (monthly)

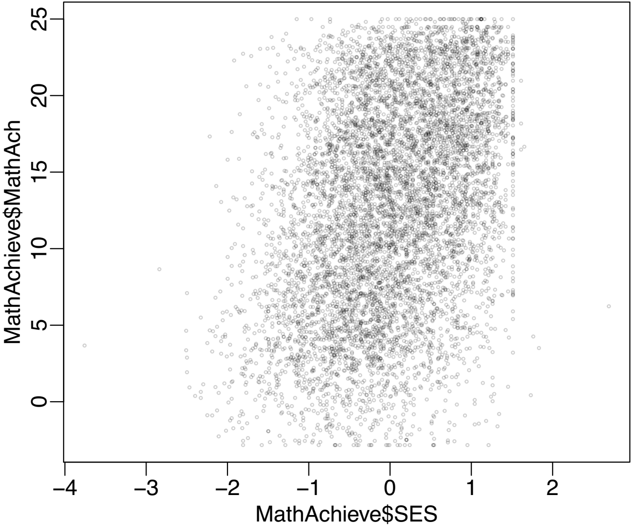
```
> plot(oDeaths$date, oDeaths$Value, xlab='date',ylab='deaths', type='l'
```



see pbrown.ca/teaching/appliedstats/data/oDeaths.R

Math scores

```
> data('MathAchieve',
+   package='MEMSS')
> plot(MathAchieve$SES,
+   MathAchieve$MathAch,
+   cex=0.3, col='#00000030')
```



Infant mortality rate

scraped from Wikipedia

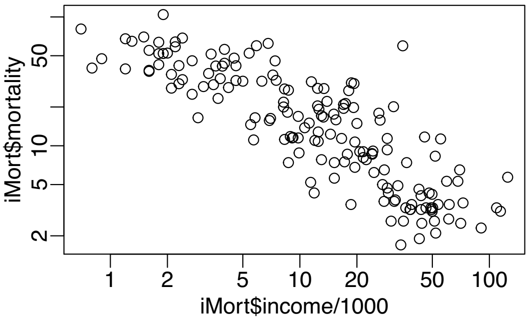
```
Warning: NAs introduced by coercion
```

```
> knitr::kable(iMort[grep("United|Japan|Egypt|Yemen|Canada",
+   iMort$Country), c("Country", "income", "gini",
+   "mortality")])
```

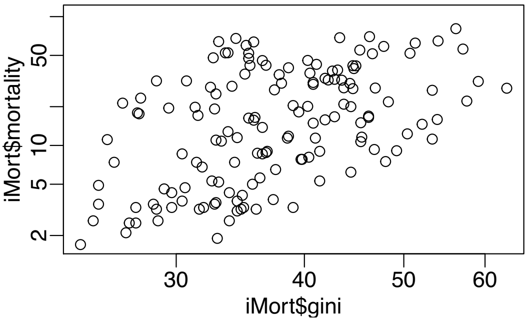
	Country	income	gini	mortality
26	Canada	48100	33.8	4.3
42	Egypt	13000	31.5	17.1
72	Japan	42700	32.9	1.9
153	United Arab Emirates	68200	32.5	5.3
154	United Kingdom	43600	34.8	4.1
155	United States	59500	41.4	5.3
160	Yemen	2300	36.7	41.9

Infant Mortality

infant mortality



income



inequality

Generalized additive models

Trevor J Hastie and Robert J Tibshirani (1990). *Generalized additive models*. Vol. 43. CRC Press

Simon Wood (2006). *Generalized additive models: an introduction with R*. CRC press

$$
\begin{aligned}
Y_i &\sim (\lambda_i, \theta) \\
g(\lambda_i) &= X_i \beta + f_1(W_{1i}) + \cdots + f_k(W_{ki})
\end{aligned}
$$

- Y_i are responses
- is the response distribution
- X_i and W_i are covariates
- $f(w)$ is some sort of wiggly line
- If we put no restrictions or assumptions on f, the estimate $\hat{f}(w)$ will interpolate the data perfectly
- ...which isn't very interesting

Penalized likelihood

$$\ell_p(\beta, f, \alpha; \mathbf{Y}) = \log[pr(\mathbf{Y}; \beta, f)] - \alpha \int \left[\left. \frac{\partial^2 f(x)}{\partial x^2} \right|_u \right]^2 du$$

$$\begin{aligned} Y_i &\sim (\lambda_i, \theta) \\ g(\lambda_i) &= X_i \beta + f(W_i) \end{aligned}$$

Estimates, given α

$$\hat{\beta}(\alpha), \hat{f}(\alpha) = \overset{\text{argmax}}{\cancel{\text{argmin}}}_{\beta, f} \ell_p(\beta, f, \alpha; \mathbf{Y})$$

- The last term is the integrated squared second derivative
- α is a penalty parameter
- A smooth $f(x)$ should have small $f"(x)$
- The second derivative is interpretable as ‘energy’
- a good $\hat{f}$ is a compromise between fitting the data and being smooth.

Cubic splines

The f which maximizes the penalized likelihod must be a cubic spline polynomial

- ...a cubic polynomial where the third derivative changes at each data point W_i
- but the 0, 1, and 2 derivatives are continuous

$$\int \left[\frac{\partial^2 f(x)}{\partial x^2}\bigg|_u \right]^2 du = F^\mathsf{T} D^\mathsf{T} R^{-1} D F$$

$f'_t \approx f_t - f_{t-1}$

$f''_t \approx f'_t - f'_{t-1}$

$\approx f_t - 2f_{t-1} + f_{t-2}$

$$F = \begin{pmatrix} f(W_1) \\ \vdots \\ \cancel{F}(W_N) \end{pmatrix} \quad D = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \end{pmatrix} \quad R = \begin{pmatrix} 2/3 & 1/6 & 0 & 0 \\ 1/6 & 2/3 & 1/6 & 0 \\ 0 & 1/6 & 2/3 & 1/6 \\ 0 & 0 & 1/6 & 2/3 \end{pmatrix}$$

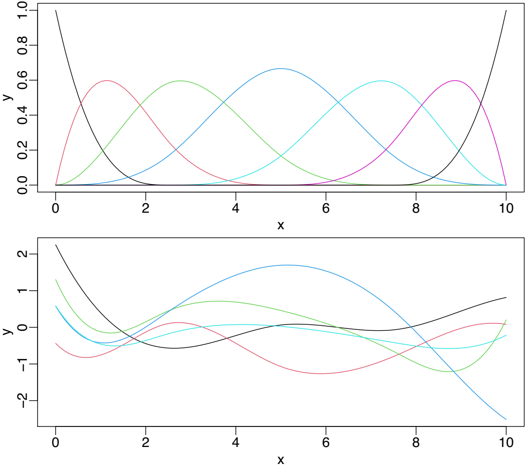
f

Peter J Green and Bernard W Silverman (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press

$F = B^T \gamma$

$\ell_p = \ln(\mathcal{L}(\beta, \gamma | Y)) - \alpha \gamma^T B D^T R^{-1} D B^T \gamma$

B-Splines



- Basis functions which approximate cubic polynomials
- There are efficient ways of maximizing the joint probability if U is restricted to B-splines

1	2	3	4	5	6	7
2.3	0.2	-1.2	0.5	-0.5	0.6	0.8
-0.4	-1.5	1.3	-1.7	-1.0	0.3	0.1
1.3	-1.0	1.0	0.5	-0.5	-1.9	0.2
0.6	-1.2	0.6	2.2	0.9	-1.8	-2.5
0.6	-1.0	0.1	0.1	-0.4	-0.7	-0.2

Paul HC Eilers and Brian D Marx (1996). “Flexible smoothing with B-splines and penalties”. In: *Statistical science*, pp. 89–102

Reduced rank approximation

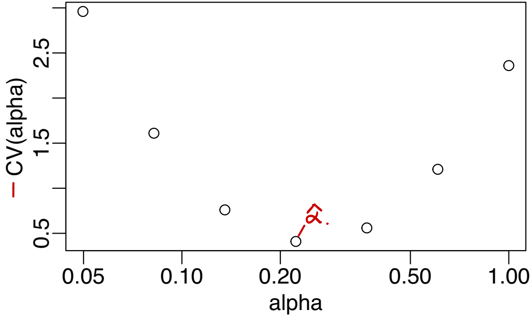
Estimating α

- Big $\alpha \Rightarrow$ smooth curve
- Small $\alpha \Rightarrow$ close fit to the data

Cross validation

- Find $\hat{\lambda}^{(-i)}$ by excluding observation i
- compute $pr(Y_i|\hat{\lambda}^{(-i)})$
- repeat for $i = 1 \dots N$
- CV$(\alpha) = \sum_i \log[pr(Y_i|\hat{\lambda}^{(-i)})]$

$$\hat{\alpha} = \text{argmax}_\alpha \text{CV}(\alpha)$$



B-splines and Generalized cross-validation

$$
\begin{aligned}
Q_{ii} &= g'(\lambda_i)^2 / \operatorname{var}(Y_i|\lambda_i) \\
B_{ij} &= B_j(W_i) \\
U &= Ba \\
B^{\mathsf{T}}Q(y-\lambda) + B^{\mathsf{T}}QBa &= (B^{\mathsf{T}}QB + \alpha D^{\mathsf{T}}D)a \\
H &= B(B^{\mathsf{T}}QB + \alpha D^{\mathsf{T}}D)^{-1}B^{\mathsf{T}}Q \\
GCV(\alpha) &= \sum_i \{(y_i - \lambda_i)^2/[N - \mathsf{trace}(H)]\}
\end{aligned}
$$

- Q depends on λ
- at iteration r, $Q^{(r)}$ is computed using $\lambda^{(r-1)}$.
- It suffices to know that the B-spline GCV algorithm is fast

Actually, the name nonparametric is not always well chosen. It might apply to kernel smoothers and running statistics, but spline smoothers are described by parameters, although their number can be large. It might be better to talk about "overparametric" techniques or "anonymous" models; the parameters have no scientific interpretation.

Paul HC Eilers and Brian D Marx (1996). "Flexible smoothing with B-splines and penalties". In: *Statistical science*, pp. 89–102

Penalized likelihood and random effects models

Random effects

$$
\begin{aligned}
Y_i|U_i &\sim (\lambda_i)\\
g(\lambda_i) &= X_i\beta + U_i\\
\mathbf{U} &\sim \text{MVN}\left[0, \frac{2}{\alpha}\left(D^\mathsf{T} R^{-1} D\right)^{-1}\right]
\end{aligned}
$$

Frequentist inference:

$$\hat{\mathbf{U}} = E(\mathbf{U}|\mathbf{Y}; \hat{\beta}, \hat{\alpha})$$

Bayesian inference:

$$\hat{\mathbf{U}} = E(\mathbf{U}|\mathbf{Y})$$

Penalized likelihood

Write $U_i = f(W_i)$

$$
\begin{aligned}
\ell_p(\beta, f, \alpha; \mathbf{Y}) &= \log[pr(\mathbf{Y}; \beta, f)] - \alpha \int f''(u)^2 du\\
&= \log[pr(\mathbf{Y}; \beta, f)] - \alpha \mathbf{U}^\mathsf{T} D^\mathsf{T} R^{-1} D \mathbf{U}\\
&= \log[pr(\mathbf{Y}, \mathbf{U}; \beta, \alpha)] + C_1\\
&= \log[pr(\mathbf{U}|\mathbf{Y}; \beta, \alpha)] + C_2
\end{aligned}
$$

Inference using

$$\hat{\mathbf{U}} = \text{argmax}_{\mathbf{U}} pr(\mathbf{U}|\mathbf{Y}; \hat{\beta}, \hat{\alpha})$$

the conditional mode

Wading into a philosophical minefield

- Frequentist:
  - plug-in MLE's of parameters α, β
  - effect f by conditional mean, or median, or mode
- Bayesian:
  - put priors on parameters α, β
  - effect f by posterior mean, or median, or mode
- Non-parametric
  - plug-in CV estimate α by CV, MLE β
  - effect f by conditional mode
  - Is this *Empirical Bayes*?
  - Restricting inference to the conditional mode is not a small detail!
  - ...the tails of the distribution are ignored

How to write a GAM

Non-parametric

$$
\begin{aligned}
Y_i \sim & (\lambda_i, \theta) \\
g(\lambda_i) = & X_i\beta + f(W_i; \nu)
\end{aligned}
$$

where $f(w;\nu)$ is a smoothly-varying function with roughness parameter ν.

Parametric

$$
\begin{aligned}
Y_i \sim & (\lambda_i, \theta) \\
g(\lambda_i) = & X_i\beta + U(W_i) \\
U(w) \sim & \text{ARIMA}_{0,2,1}(\sigma^2, 2 - \sqrt{3})
\end{aligned}
$$

$U(w)$ is ~~a~~ almost random walk of order 2 with variance σ^2.

Patrick E Brown and P de Jong (2001). "Nonparametric smoothing using state space techniques". In: *Canadian Journal of Statistics* 29.1, pp. 37–50. DOI: 10.2307/3316049

Math

$Y_i \sim N(\mu_i, \tau^2)$

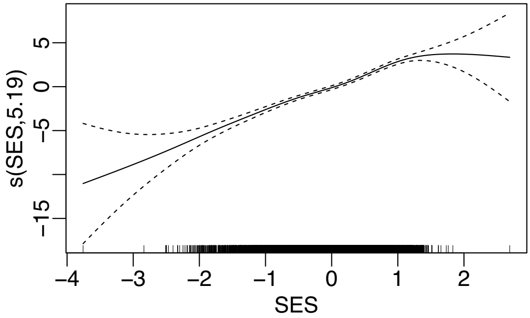
$\mu_i = X_i\beta + f(W_i)$

SES.

```
> library('mgcv')
> mathGam = gam(
+   MathAch ~ s(SES) + Minority*Sex,
+   data=MathAchieve)
> knitr::kable(
+   summary(mathGam)$p.table[,1:2],
+   digits=1)
```

	Estimate	Std. Error
(Intercept)	12.9	0.1
MinorityYes	-2.7	0.2
SexMale	1.4	0.2
MinorityYes:SexMale	-0.2	0.3

```
> plot(mathGam)
```




```
> mathGam$sp
    s(SES)
0.8254378
```

$\hat{\alpha}$

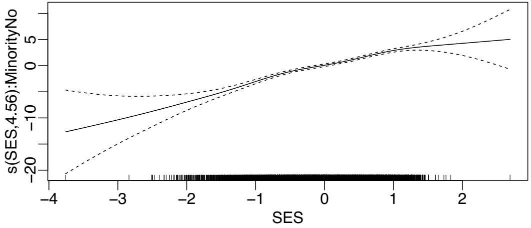
$$\mu_i = X_i\beta + \begin{cases} f_1(W_i), & X_{i1}=1 \\ f_2(W_i), & X_{i1}=0 \end{cases}$$

Math, SES-minority interaction

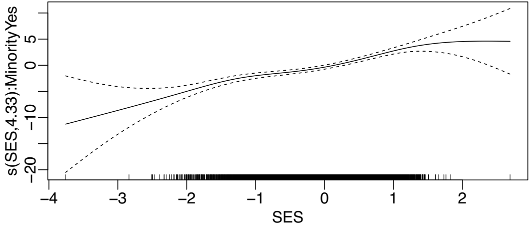
```
> mathGamInt = gam(
+   MathAch ~ s(SES, by=Minority) +
+     Minority*Sex,
+   data=MathAchieve)
> mathGamInt$sp
 s(SES):MinorityNo s(SES):MinorityYes
          0.820158           0.614983
```

	Estimate	Std. Error
(Intercept)	12.8	0.1
MinorityYes	-2.9	0.2
SexMale	1.4	0.2
MinorityYes:SexMale	-0.1	0.3

```
> plot(mathGamInt, select = 1)
```




```
> plot(mathGamInt, select = 2)
```



A common smoothing parameter?

$$
\begin{aligned}
Y_{ij} &\sim N(\lambda_{ij}, \tau^2) \\
\lambda_{ij} &= X_{ij}\beta + f_i(W_{ij};\nu)
\end{aligned}
$$

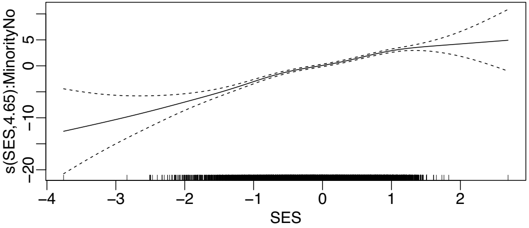
- Y_{ij} is the observation for individual j in ethnic group i
- X_{ij} is a vector of covariates (ethnic group, sex, interaction)
- $f_i(w;\nu)$ is the smoothly-varying function of SES
  - for ethnic group i
  - with roughness parameter ν.

Math, common smoothing parameter

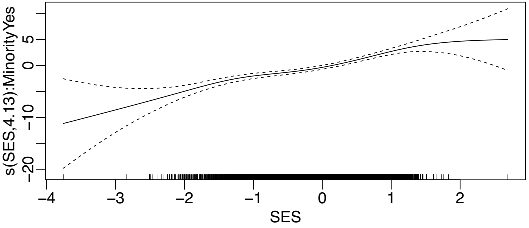
```
> mathGamIntC = gam(MathAch ~
+   s(SES, by=Minority, id=1) +
+     Minority*Sex,
+   data=MathAchieve)
> mathGamIntC$sp
s(SES):MinorityNo
        0.7492505
```

	Estimate	Std. Error
(Intercept)	12.8	0.1
MinorityYes	-2.9	0.2
SexMale	1.4	0.2
MinorityYes:SexMale	-0.1	0.3

```
> plot(mathGamIntC, select = 1)
```




```
> plot(mathGamIntC, select = 2)
```



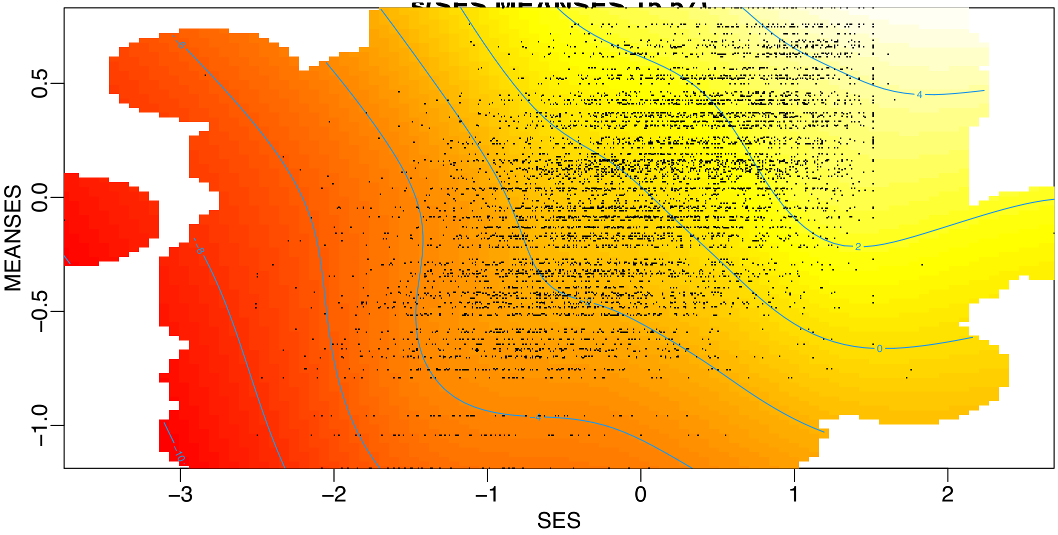
Math 2d

- two dimensional smoothing
- penalize curvature
- 2d b-splines

school average

```
> mathGam2 = gam(MathAch ~ s(SES, MEANSES) + Minority *
+   Sex, data = MathAchieve)
> plot(mathGam2, scheme = 2, n2 = 100)
```

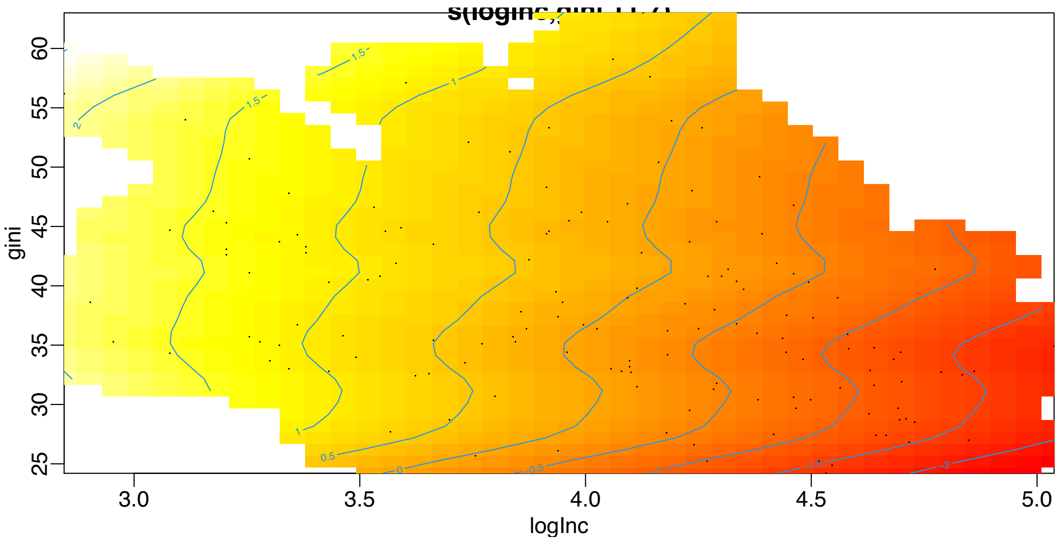
Math 2d



Infant mortality

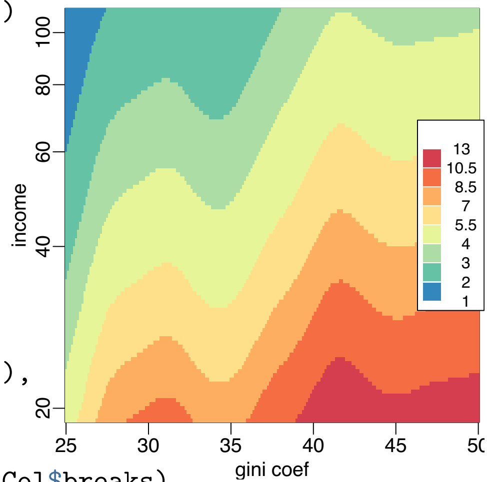
```
> library("mgcv")
> iMort$logInc = log10(iMort$income)
> iMort$logMort = log(iMort$mortality)
> mortGam = gam(logMort ~ s(logInc, gini), data = iMort)
> plot(mortGam, scheme = 2)
```

Infant mortality



Infant mortality

```
> predList = list(gini = seq(25,50,len=201), logInc = seq(
+   log10(19000), log10(110000), len=101))
> mortPred = exp(predict(mortGam,
+   do.call(expand.grid, predList),
+   type='response'))
> mortCol = mapmisc::colourScale(
+   mortPred, digits=1.5, col='Spectral',
+   style='equal', transform=0.5,
+   breaks=9, rev=TRUE)
> image(predList$gini,
+   10^predList$logInc/1000,
+   matrix(mortPred, length(predList$gini),
+     length(predList$logInc)),
+   xlab = 'gini coef', ylab='income',
+   log='y', col=mortCol$col, breaks=mortCol$breaks)
> mapmisc::legendBreaks("right", mortCol, cex=0.8, inset=0)
```



Deaths in Ontario

```
> timeOrigin = ISOdate(2000,1,1,0,0,0, tz='UTC')
> oDeaths$timeNumeric = as.numeric(
+     difftime(oDeaths$date, timeOrigin,
+     units='days'))
> oDeaths[c(1,100,200),
+   c('date', 'month', 'Value',
+     'timeNumeric')]
           date    month Value timeNumeric
159  1991-01-01 January  6542       -3287
3077 1999-04-01    April  6463        -275
6231 2007-08-01   August  6527        2769
```

- Monthly death count is `Value`
- The `date` variable is a fancy data type for times
- `strptime` creates date objects
- `timeNumeric` is number of days since 1 Jan 2000

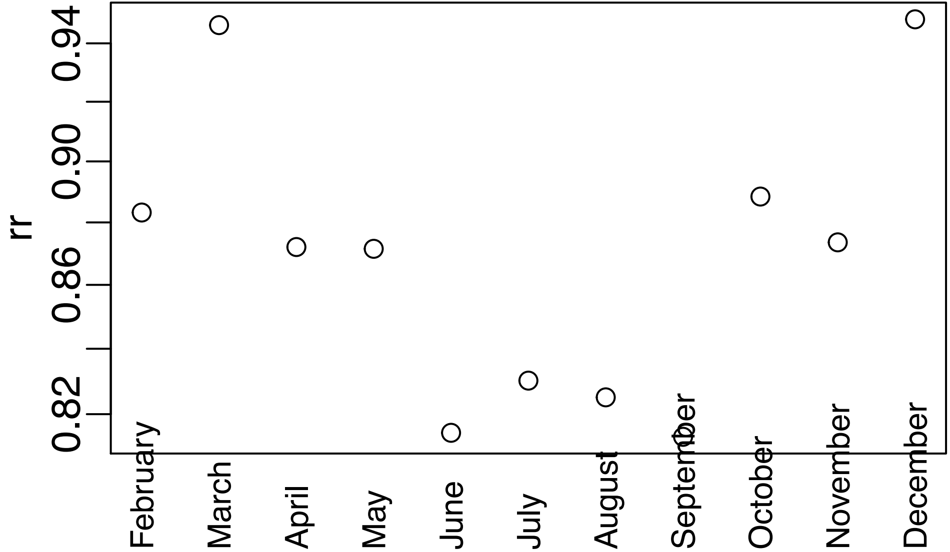
Ontario GAM

$$
\begin{aligned}
Y_i \sim& \text{Poisson}(\lambda_i) \\
\log(\lambda_i) =& X_i\beta + f(time_i)
\end{aligned}
$$

```
> deathsGam = gam(
+   Value ~ month + s(timeNumeric),
+   data=oDeaths, family='poisson'
+ )
> knitr::kable(
+   summary(deathsGam)$p.table[,1:2],
+   digits=3, col.names=c('est','se'))
```

	est	se
(Intercept)	9.001	0.002
monthFebruary	-0.124	0.003
monthMarch	-0.055	0.003
monthApril	-0.137	0.003
monthMay	-0.138	0.003
monthJune	-0.205	0.003
monthJuly	-0.186	0.003
monthAugust	-0.192	0.003
monthSeptember	-0.207	0.003
monthOctober	-0.118	0.003
monthNovember	-0.135	0.003
monthDecember	-0.053	0.003

Relative rate for each month



Number of days in each month

```
> oDeaths$daysInMonth = Hmisc::monthDays(oDeaths$date)
> oDeaths$nDays = log(oDeaths$daysInMonth)
> oDeaths[c(1, 2, 100, 200), c("date", "month",
+   "daysInMonth", "nDays")]
```

```
           date    month daysInMonth    nDays
159  1991-01-01  January          31 3.433987
161  1991-02-01 February          28 3.332205
3077 1999-04-01    April          30 3.401197
6231 2007-08-01   August          31 3.433987
```

- calculate the number of days in each month
- the log number of days will be an offset
- rates modelled will be number of deaths per day

Ontario GAM with offset

$$
\begin{aligned}
Y_i \sim & \text{Poisson}(O_i\lambda_i) \\
\log(\lambda_i) = & X_i\beta + f(time_i)
\end{aligned}
$$

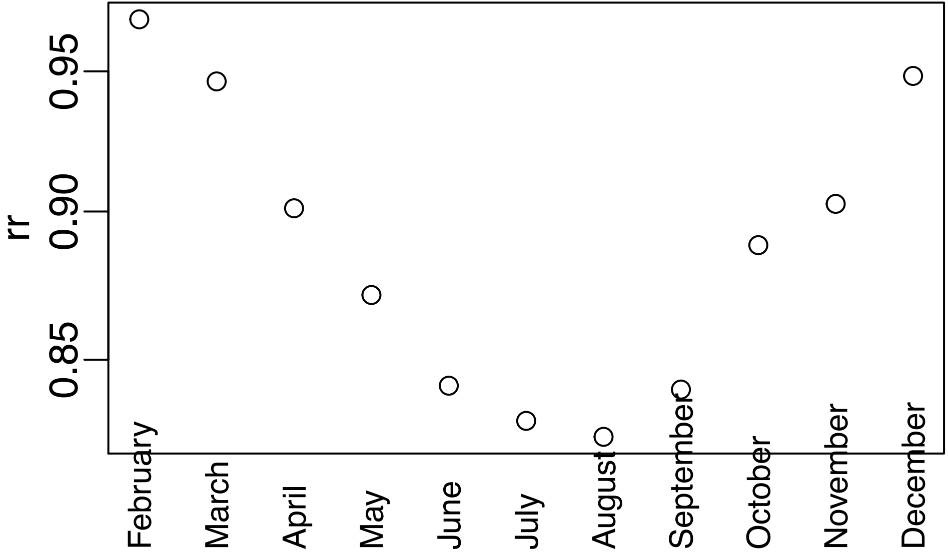
O_i: days in month

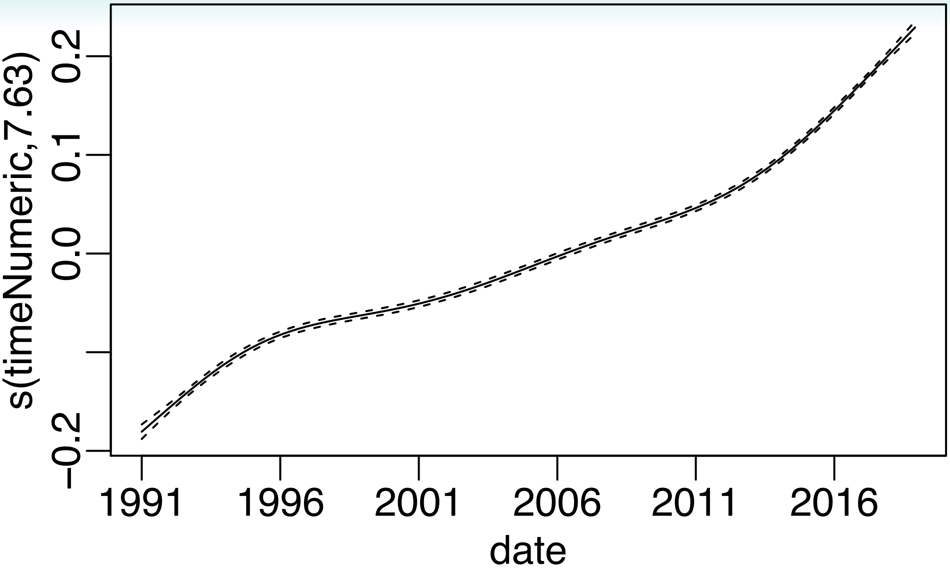
λ_i: deaths / day

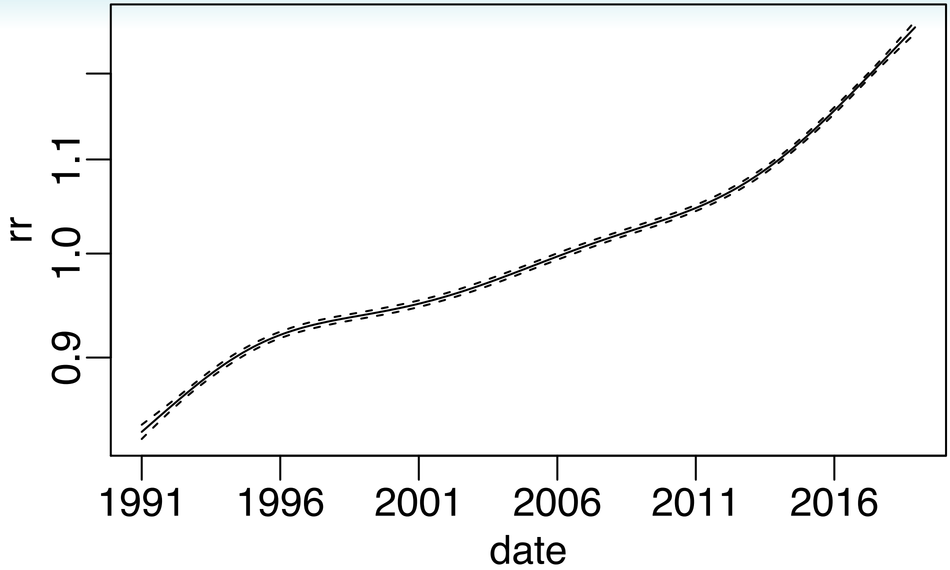
```
> deathsGam = gam(
+   Value ~ month + s(timeNumeric) +
+     offset(nDays), data=oDeaths,
+   family='poisson')
> knitr::kable(
+   summary(deathsGam)$p.table[,1:2],
+   digits=3, col.names=c('est','se'))
```

	est	se
(Intercept)	5.567	0.002
monthFebruary	-0.031	0.003
monthMarch	-0.055	0.003
monthApril	-0.104	0.003
monthMay	-0.138	0.003
monthJune	-0.173	0.003
monthJuly	-0.186	0.003
monthAugust	-0.192	0.003
monthSeptember	-0.174	0.003
monthOctober	-0.118	0.003
monthNovember	-0.102	0.003
monthDecember	-0.053	0.003

Relative rate for each month




s(timeNumeric,7.63)
−0.2
0.0
0.1
0.2
1991
1996
2001
2006
2011
2016
date


rr
0.9
1.0
1.1
1991
1996
2001
2006
2011
2016
date

```
> dSeq = seq(from = min(oDeaths$date), by = "5 years",
+   length.out = 10)
> deathPred = as.matrix(as.data.frame(predict.gam(deathsGam,
+   oDeaths, type = "terms", terms = "s(timeNumeric)",
+   se.fit = TRUE)))
> deathPred = exp(deathPred %*% Pmisc::ciMat())
> matplot(oDeaths$timeNumeric, deathPred, log = "y",
+   xaxt = "n", xlab = "date", type = "l", lty = c(1,
+     2, 2), col = "black", ylab = "rr")
> axis(1, at = difftime(dSeq, timeOrigin, units = "days"),
+   labels = format(dSeq, "%Y"))
```

Forecasting

```
> newX = data.frame(date = seq(from = timeOrigin,
+   by = "months", length.out = 12 * 21))
> newX$timeNumeric = as.numeric(difftime(newX$date,
+   timeOrigin, units = "days"))
> newX$daysInMonth = Hmisc::monthDays(newX$date)
> newX$nDays = log(newX$daysInMonth)
> newX$month = months(newX$date)
```

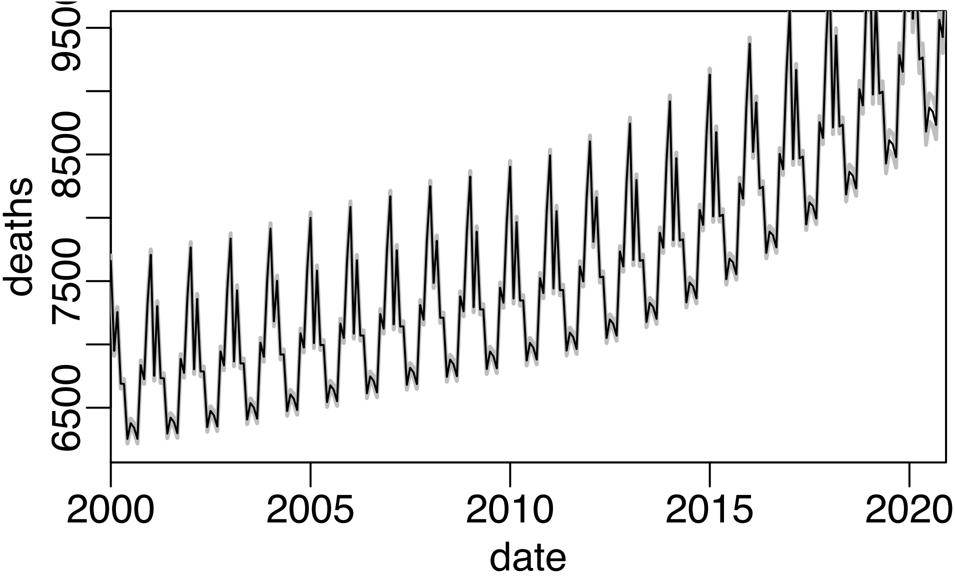
Predictions

```
> deathsPred = predict(deathsGam, newX, se.fit = TRUE)
> deathsPred = cbind(newX, deathsPred)
> deathsPred$lower = deathsPred$fit - 2 * deathsPred$se.fit
> deathsPred$upper = deathsPred$fit + 2 * deathsPred$se.fit
> for (D in c("fit", "lower", "upper")) {
+   deathsPred[[paste(D, "exp", sep = "")]] = exp(deathsPred[[D]])
+ }
```

Forecasts

```
> plot(deathsPred$date, deathsPred[, "fitexp"],
+   type = "n", xlab = "date", ylab = "deaths",
+   xaxs = "i", ylim = c(6200, 9500))
>
> matlines(deathsPred$date, deathsPred[, c("lowerexp",
+   "upperexp", "fitexp")], lty = 1, col = c("grey",
+   "grey", "black"), lwd = c(2, 2, 1))
> points(oDeaths$date, oDeaths$Value, cex = 0.5,
+   col = "red")
```

Forecasts



Favoured models

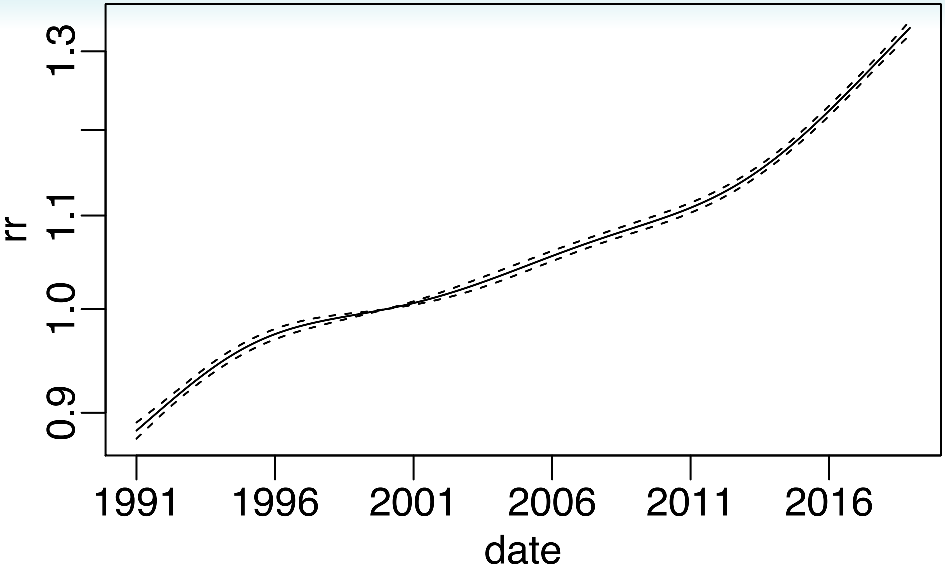
- Nancy E. Heckman and James O. Ramsay (2000). "Penalized Regression with Model-Based Penalties". In: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 28.2, pp. 241–258. URL: http://www.jstor.org/stable/3315976
- what happens when $\alpha \to \infty$?
- what f has zero penalty?
- straight lines have $f''(x) = 0$
- PLS with second derivative penalty encourages straight lines
- connections with Simpson et al. (2017)
- using PLS implicitly assumes f is almost straight

a different constraint

- adding a constant to $f(x)$ doesn't change the penalty
- … any straight line has penalty zero
- can't identify both $f(x)$ and the intercept without constraints
- a sum-to-zero constraint on $f(x)$ is the default
- … alternative is to force $f(x_0) = 0$

```
> deathsGamC = gam(
+   Value ~ month + s(timeNumeric, pc=0) +
+     offset(nDays), data=oDeaths,
+   family='poisson')
```

	est	se
(Intercept)	5.510	0.003
monthFebruary	-0.031	0.003
monthMarch	-0.055	0.003
monthApril	-0.104	0.003
monthMay	-0.138	0.003
monthJune	-0.173	0.003
monthJuly	-0.186	0.003
monthAugust	-0.192	0.003
monthSeptember	-0.174	0.003
monthOctober	-0.118	0.003
monthNovember	-0.102	0.003
monthDecember	-0.053	0.003


1.3
1.1
1.0
0.9
rr
1991
1996
2001
2006
2011
2016
date

```r
> deathPredC = as.matrix(as.data.frame(predict.gam(deathsGamC,
+   oDeaths, type = "terms", terms = "s(timeNumeric)",
+   se.fit = TRUE)))
> deathPredC = exp(deathPredC %*% Pmisc::ciMat())
> matplot(oDeaths$timeNumeric, deathPredC, log = "y",
+   xaxt = "n", xlab = "date", type = "l", lty = c(1,
+     2, 2), col = "black", ylab = "rr")
> axis(1, at = difftime(dSeq, timeOrigin, units = "days"),
+   labels = format(dSeq, "%Y"))
```

About GAM's

- They're easy to fit
- ...but inference is slightly unusual
  - Maximizing the joint probability $pr(Y, U; \beta, \alpha)$
  - penalty from cross validation

Bayesian inference

- put priors on α, β
- and compute posteriors
- ...with INLA (of course)

Model for Ontario deaths

$$
\begin{aligned}
Y_i \sim& \text{Poisson}(O_i \lambda_i)\\
\log(\lambda_i) =& X_i \beta + U(t_i) + V_i\\
[U_1 \ldots U_T]^T \sim& \text{RW2}(0, \sigma_U^2)\\
V_i \sim& \text{N}(0, \sigma_V^2) \quad \text{(independent)}
\end{aligned}
$$

- $U(t)$ is a second-order random walk
- ...second derivatives are $\text{N}(0, \sigma_U^2)$
- V_i covers independent variation or over-dispersion

Random Walks

- RW(0), independent

$$V_t \sim \text{iid N}(0, \tau^2)$$

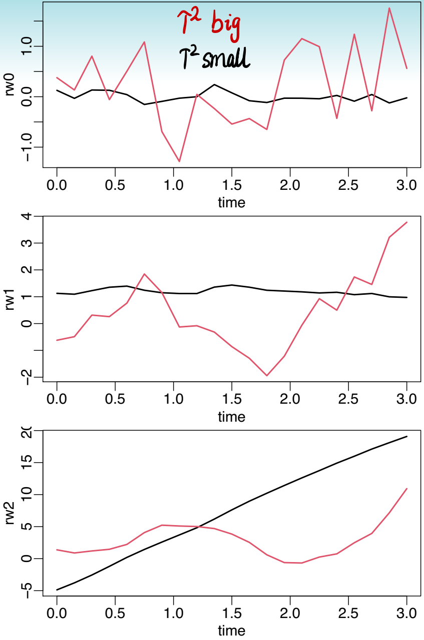
- RW(1), Brownian motion

$$V_{t+1} | V_k, k < t \sim \text{N}(V_t, \tau^2)$$
$$V_{t+1} - V_t \sim \text{N}(0, \tau^2)$$

- RW(2), Random slope

$$V_{t+1} | V_k, k < t \sim \text{N}(-2V_t + V_{t-1}, \tau^2)$$
$$(V_{t+1} - V_t) - (V_t - V_{t-1}) \sim \text{N}(0, \tau^2)$$
$$V_{t+1} - 2V_t + V_{t-1} \sim \text{N}(0, \tau^2)$$

$$V_{t+1} - V_t \sim N(V_t - V_{t-1}, \tau^2)$$



Favoured models and constraints

- $\alpha = \infty$ equivalent to $\overset{\tau}{\cancel{\alpha}} = 0$
  - base model, favoured model
- RW(0): $V_t = 0$ for all t
- RW(1): $V_t = V_0$, horizontal line
- RW(2): $V_t = V_0 + t \cdot (V_1 - V_0)$, straight line
- RW(1) or RW(2)? favour straight line or horizontal line?
- PC prior: how strongly to encourage favoured model

Preparing the date variable

- there are many ways to convert dates to numbers
- don't rely on `inla` to do the right thing
- R's `Date` objects are stored as number of days since 1 January 1970
- convert time from days to years, numerically more stable

```
> class(oDeaths$date)
```

```
[1] "Date"
```

```
> oDeaths$date[1:4]
```

```
[1] "1991-01-01" "1991-02-01" "1991-03-01" "1991-04-01"
```

```
> as.numeric(oDeaths$date)[1:4]
```

```
[1] 7670 7701 7729 7760
```

```
> oDeaths$timeIid = oDeaths$timeRw = as.numeric(oDeaths$date)/365.25
> oDeaths$timeRw[1:4]
```

Running INLA

- Prior for RW: slope of log rate changes by 0.2 from one year to the next
- `scale.model` defaults to `TRUE`, which multplies σ by a constant that some people (not me) find more interpretable.

```
> library("INLA")
> deathsBayes = inla(
+     Value ~ offset(nDays) + month +
+     f(timeRw, model='rw2',
+       prior='pc.prec', param=c(0.2, 0.5),
+       scale.model=FALSE) +
+    f(timeIid, model='iid',
+       prior='pc.prec', param=c(log(1.25), 0.5)),
+     data=oDeaths, family='poisson',
+     control.predictor=list(compute=TRUE))
```

(small)

25% change is the prior median (a lot).

$Y_i \sim \text{Poisson}(O_i \lambda_i)$

$\log(\lambda_i) = X_i\beta + U_i + V_i$

$V_i \overset{\text{independent}}{\sim} N(0, \tau^2)$

$U_i \sim RW2(\sigma^2)$

Priors:

$\beta \sim MVN(0, KI)$

$\tau \sim \text{Exp}$, median $\log(1.25)$

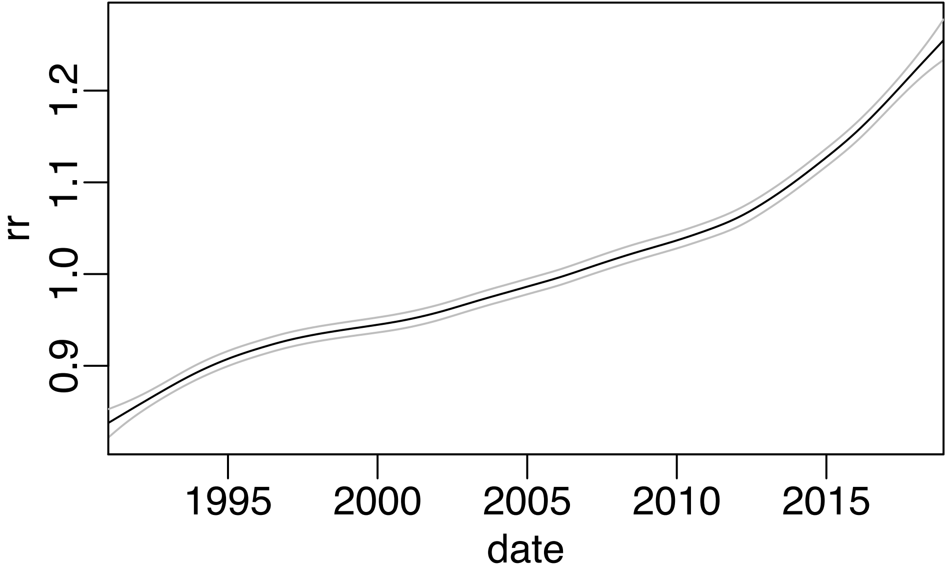
$\sigma \sim \text{Exp}$, median 0.2

Parameters

```
> Squant = paste0(c(0.5, 0.025, 0.975), "quant")
> knitr::kable(deathsBayes$summary.fixed[, Squant],
+   digits = 4)
```

	0.5quant	0.025quant	0.975quant
(Intercept)	5.5644	5.5530	5.5757
monthFebruary	-0.0299	-0.0461	-0.0138
monthMarch	-0.0528	-0.0689	-0.0367
monthApril	-0.1024	-0.1186	-0.0863
monthMay	-0.1350	-0.1511	-0.1189
monthJune	-0.1697	-0.1859	-0.1535
monthJuly	-0.1833	-0.1994	-0.1671
monthAugust	-0.1902	-0.2064	-0.1740
monthSeptember	-0.1719	-0.1881	-0.1557
monthOctober	-0.1159	-0.1320	-0.0997
monthNovember	-0.1002	-0.1164	-0.0841

Time effect $[U_i | Y]$



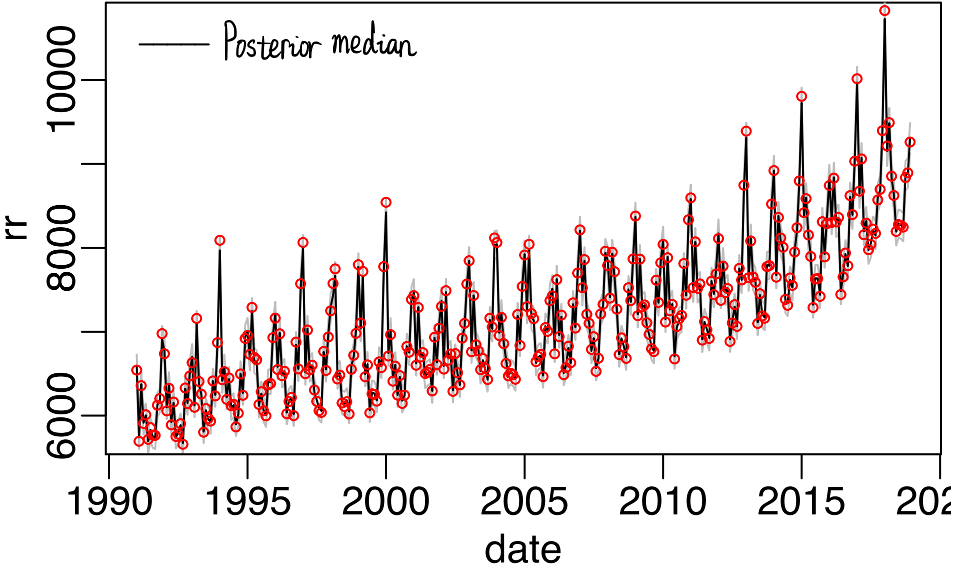
Time effect

X, in years since 1970

```
> matplot(deathsBayes$summary.random$timeRw$ID,
+   exp(deathsBayes$summary.random$timeRw[, c("0.025quant",
+     "0.975quant", "0.5quant")]), xaxt = "n",
+   xlab = "date", lty = 1, col = c("grey", "grey",
+     "black"), type = "l", ylab = "rr", xaxs = "i")
> Saxis = seq(as.Date("1990/1/1"), as.Date("2025/1/1"),
+   by = "5 years")
> axis(1, at = as.numeric(Saxis)/365.25, labels = format(Saxis,
+   "%Y"))
```

Rate (deaths per month)

~~Fitted values~~ $[O_i \lambda_i | Y]$



Fitted values

$E(Y_i) = O_i \lambda_i$

```
> matplot(oDeaths$date, deathsBayes$summary.fitted[,
+   c("0.025quant", "0.975quant", "mean")], xaxt = "n",
+   xlab = "date", lty = 1, col = c("grey", "grey",
+     "black"), type = "l", ylab = "rr", yaxs = "i")
> points(oDeaths$date, oDeaths$Value, cex = 0.5,
+   col = "red")
> axis(1, at = Saxis, labels = format(Saxis, "%Y"))
```

Parameters

```
> deathsBayes$priorPost = Pmisc::priorPostSd(deathsBayes)
> knitr::kable(deathsBayes$priorPost$summary[, Squant],
+   digits = 4)
```

		0.5quant	0.025quant	0.975quant
σ	SD for timeRw	0.0050	0.0026	0.0103
τ	SD for timeIid	0.0283	0.0258	0.0311

- How do we interpet Sd for timeRw?
- very hard, most people don't
- change in slope over one year, not surprising it's small

Re-parametrized

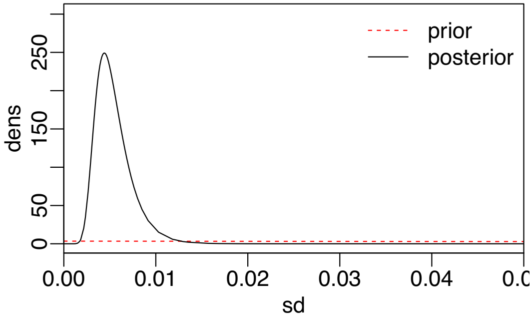
```
> toPrint = diag(c(sqrt(25),1)) %*%
+   as.matrix(deathsBayes$priorPost$summary[,
+     c('mean','0.025quant','0.975quant')])
> rownames(toPrint) = c('SD RW2, 25 yr', 'SD independent')
> knitr::kable(toPrint, digits=3)
```

$\sqrt{25\sigma^2}$

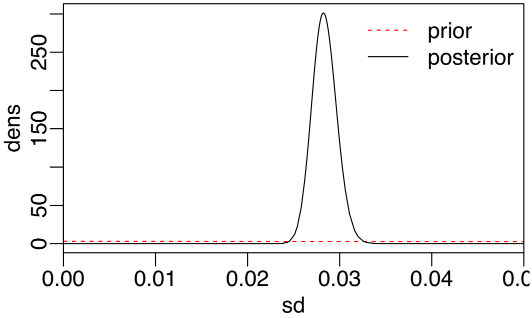
T

	mean	0.025quant	0.975quant
SD RW2, 25 yr	0.027	0.013	0.052
SD independent	0.028	0.026	0.031

sd prior, post



rw2 σ



iid τ

prior, post

```
> deathsBayes$priorPost[[1]]$matplot$ylim =
+  deathsBayes$priorPost[[2]]$matplot$ylim =
+   range(deathsBayes$priorPost[[2]]$matplot$y)
> for(Dparam  in 1:2){
+   do.call(matplot,deathsBayes$priorPost[[Dparam]]$matplot)
+   do.call(legend,deathsBayes$priorPost$legend)
+ }
```

Telling INLA we want forecasts

```
> newX = data.frame(
+   date = seq(max(oDeaths$date),
+     as.Date('2025/1/1'), by='month')[-1],
+    Value = NA)
> newX$timeIid = newX$timeRw  =
+   as.numeric(newX$date)/365.25
> newX$daysInMonth = Hmisc::monthDays(newX$date)
> newX$nDays = log(newX$daysInMonth)
> newX$month  = months(newX$date)
>
> newData = rbind(oDeaths[,names(newX)], newX)
```

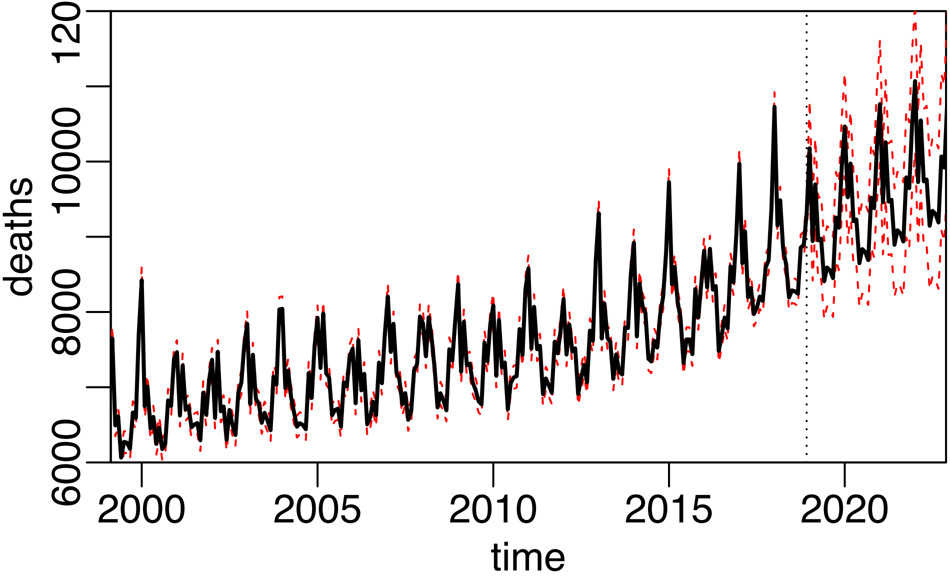
- Create a sequence of dates in a new data frame
- missing values for observed deaths
- convert dates to the values `inla` wants
- days in each month and name of month
- combine with the original data
- could have used `lincombs` instead

Re-runing with these new data

```
> deathsBayesRefit = inla(
+   deathsBayes$.args$formula,
+     data=newData, family='poisson',
+     control.predictor=list(compute=TRUE, link=1))
```

- `link=1` is a bizarre `inla` requirement
- it means use the link function from the first family
- ... note there's only one family

Predicted values



Predicted values

```
> deathsPred = deathsBayesRefit$summary.fitted.values
> theXlim = as.Date(c("2000/1/1", "2022/1/1"))
> matplot(newData$date, deathsPred[, c("0.025quant",
+   "0.975quant", "0.5quant")], xlab = "time",
+   ylab = "deaths", xlim = theXlim, lwd = c(1,
+     1, 2), ylim = c(6000, 12000), type = "l",
+   lty = c(2, 2, 1), col = c("red", "red", "black"),
+   xaxt = "n", yaxs = "i")
> abline(v = as.numeric(max(oDeaths$date)), lty = 3)
> axis(1, at = Saxis, labels = format(Saxis, "%Y"))
```

Posterior samples

$$
\begin{aligned}
Y_i \sim& \text{Poisson}(O_i \lambda_i) \\
\log(\lambda_i) =& X_i \beta + U(t_i) + V_i \\
[U_1 \ldots U_T]^T \sim& \text{RW2}(0, \sigma_U^2) \\
V_i \sim& \text{N}(0, \sigma_V^2)
\end{aligned}
$$

- we've been plotting the posterior medians and quantiles of $\pi(U|Y)$
- we can take random samples of from the posterior of U
- ... and plot those
- the method for doing this in `inla` isn't published, and is based on a Normal approximation.
- MCMC is an alternative with some advantages for this task

Recompute

```
> deathsBayesC = inla(deathsBayes$.args$formula,
+     data=newData, family='poisson',
+     control.compute = list(config=TRUE),
+     control.predictor=list(compute=TRUE, link=1))
> print(object.size(deathsBayesRefit), units='MB')
3.4 Mb
> print(object.size(deathsBayesC), units='MB')
16.4 Mb
```

- config means save a load of variance matrices
- which results in the new fitted object being much larger

Posterior samples

- I want 64 samples from the posterior $\pi(U(t)|Y)$
- run ~~16~~ 8 threads in parallel, ~~4~~ 8 samples per thread
- give me only the `timeRw` part, at all time points

```
> mySampleList = inla.posterior.sample(n = 64, result = deathsBayesC,
+   num.threads = 8, selection = list(timeRw = seq(1,
+     nrow(deathsBayesC$summary.random$timeRw))))
```

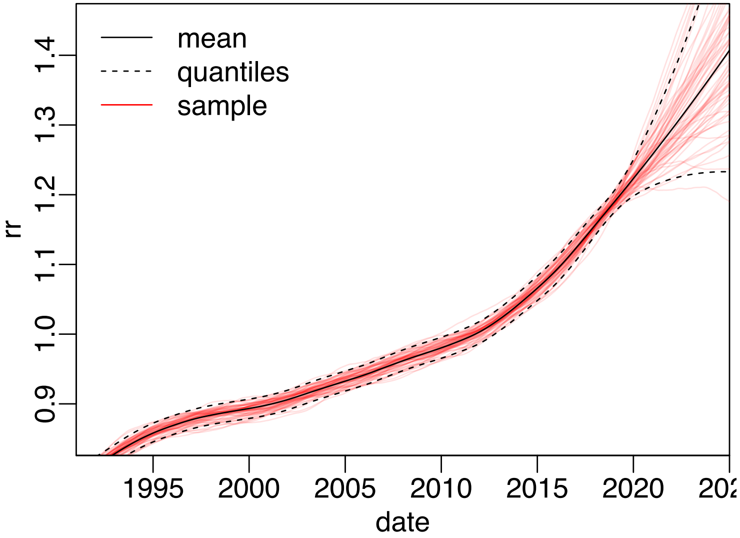
```
> length(mySampleList)
```

```
[1] 64
```

```
> mySample = do.call(cbind, Biobase::subListExtract(mySampleList,
+   "latent"))
> dim(mySample)
```

```
[1] 409  64
```

Plot of posterior sample [U|Y]



- samples are semi-transparent
- intervals are *pointwise*
- at each time point, interval contains 95% of samples
- can we get an envelope which contains 95% of samples?

Plot of posterior sample

```
> matplot(newData$date, exp(mySample), xaxt = "n",
+   xlab = "date", lty = 1, col = "#FF000020",
+   ylim = c(0.85, 1.45), type = "l", ylab = "rr",
+   xaxs = "i")
> matlines(newData$date, exp(deathsBayesRefit$summary.random$timeRw[,
+   Squant]), lty = c(1, 2, 2), col = "black")
> axis(1, at = Saxis, labels = format(Saxis, "%Y"))
> legend("topleft", bty = "n", lty = c(1, 2, 1),
+   col = c("black", "black", "red"), legend = c("mean",
+     "quantiles", "sample"))
```

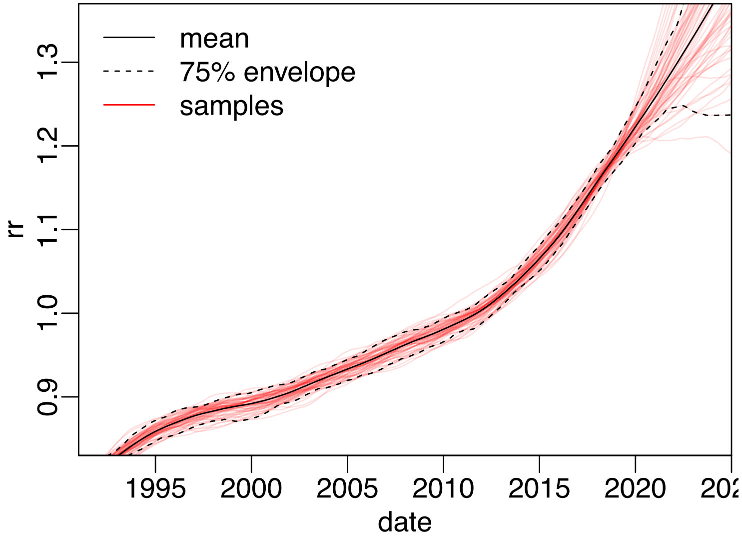
Joint confidence regions

```
> cset <- GET::create_curve_set(list(
+   r = as.numeric(newData$date),
+   obs = mySample))
> myEnv = GET::central_region(cset,
+   coverage=0.75)
```

- global envelopes
- the GET package
- Myllymäki et al. (2017)
- this is *functional data analysis*, an active research topic

Mari Myllymäki et al. (2017). "Global envelope tests for spatial processes". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.2, pp. 381–404. DOI: 10.1111/rssb.12172

Plot of posterior sample



- intervals are *global*
- 3/4 of samples are in the interval
- whatever the 'true' trend is, there's a 75% chance it's entirely contained within this envelope

Plot of posterior sample

```
> matplot(newData$date, exp(mySample), xaxt = "n",
+   xlab = "date", lty = 1, col = "#FF000020",
+   ylim = c(0.85, 1.35), type = "l", ylab = "rr",
+   xaxs = "i")
> matlines(newData$date, exp(as.data.frame(myEnv)[,
+   c("lo", "hi", "central")]), lty = c(2, 2,
+   1), col = "black")
> axis(1, at = Saxis, labels = format(Saxis, "%Y"))
> legend("topleft", bty = "n", lty = c(1, 2, 1),
+   col = c("black", "black", "red"), legend = c("mean",
+     "75% envelope", "samples"))
```

let's revisit the RW2 SD

```
> secondDiff = apply(mySample, 2,
+   diff, differences=2)
> dim(secondDiff)
[1] 407  64
> matplot(newData$date[-1],
+   secondDiff, type='l',
+   col='#FF000020', lty=1)
```

```
Error in matplot(newData$date[-1], sec
```

1 year differences

```
> quantile(apply(secondDiff, 2, sd))

          0%          25%          50%          75%         100%
9.065556e-05 1.297998e-04 1.396620e-04 1.708918e-04 2.742733e-04

> signif(sqrt(12) * quantile(apply(secondDiff, 2,
+   sd)), 4)

       0%       25%       50%       75%      100%
0.0003140 0.0004496 0.0004838 0.0005920 0.0009501

> Pmisc::priorPostSd(deathsBayes)$summary[1, Squant]

                  0.5quant 0.025quant 0.975quant
SD for timeRw 0.005047952 0.00264209 0.01030015
```

- sd of secondDiff is monthly sd
- yearly var is 12 times monthly var
- agrees with parameter estimates

Notes on INLA/Bayes

- INLA fit an RW2 model,
- not quite the same as the ARIMA(0,2,1)
- and not a cubic spline
- so INLA is rougher, second derivatives are independent, not continuous

Math

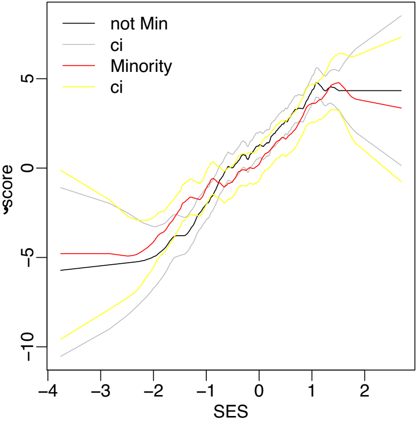
$$
\begin{aligned}
Y_{ijk} &\sim \mathrm{N}(\mu_{ijk}, \tau^2) \\
\mu_{ijk} &= X_{ijk}\beta + U_i + f_j(W_{ijk}) \\
U_i &\sim \mathrm{N}(0, \sigma_U^2) \\
f_j(s+1) - f_j(s) &\sim \mathrm{N}(0, \sigma_f^2)
\end{aligned}
$$

- school i, ethnicity j, individual k
- W_{ijk} is the SES variable
- non-parameteric SES effect
  - `f(SES, model='rw1')`
- SES/ethnicity interaction
  - `f(SES, model='rw1', replicate = ethnicity)`

Math

```
> MathAchieve$minorityNum = as.numeric(MathAchieve$Minority)
> mathInla = inla(MathAch~f(SES,
+       replicate=minorityNum,
+       model='rw1', prior='pc.prec',
+       param=c(1, 0.5), scale.model=FALSE
+    ) + Minority*Sex,
+    control.family = list(hyper = list(
+ prec=list(prior='pc.prec',param=c(1,0.5)))),
+    data=MathAchieve)
```

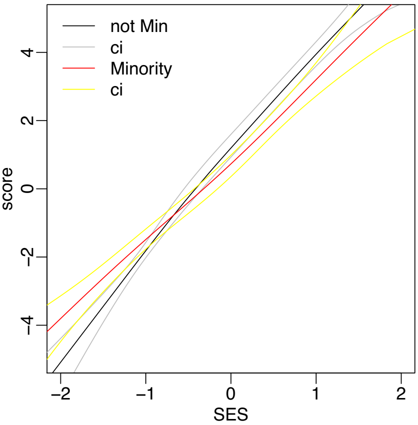
	mean	2.5	97.5
(Intercept)	11.7	11.3	12.0
MinorityYes	-2.5	-3.0	-1.9
SexMale	1.4	1.1	1.7
MinorityYes:SexMale	-0.1	-0.8	0.5



Math, rw2

```
> mathInla2 = inla(
+   MathAch ~ f(SES, replicate=minorityNum,
+           model = 'rw2', prior='pc.prec',
+      param=c(0.2, 0.5), scale.model=FALSE
+      ) + Minority*Sex,
+      control.inla = list(h=0.00001),
+     control.family = list(hyper = list(prec=
+ list(prior='pc.prec', param = c(1, 0.5)))),
+   data=MathAchieve)
```

	mean	2.5	97.5
(Intercept)	11.7	11.4	12.0
MinorityYes	-2.5	-3.1	-2.0
SexMale	1.4	1.1	1.7
MinorityYes:SexMale	-0.1	-0.8	0.5

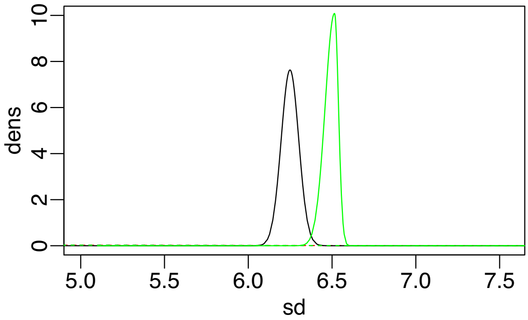


Parameters

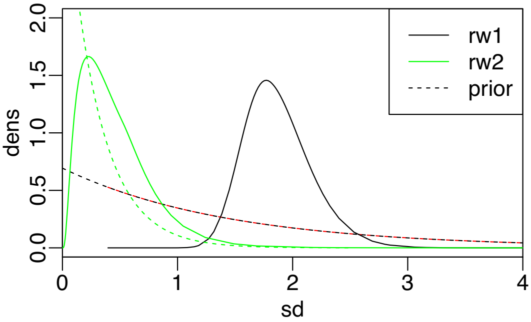
```
> mathInla$priorPost = Pmisc::priorPost(mathInla)
> mathInla2$priorPost = Pmisc::priorPost(mathInla2)
>
> knitr::kable(rbind(mathInla$priorPost[[1]]$summary[,
+   Squant], mathInla2$priorPost[[1]]$summary[,
+   Squant], mathInla$priorPost[[2]]$summary[,
+   Squant], mathInla2$priorPost[[2]]$summary[,
+   Squant]), digits = 5)
```

	0.5quant	0.025quant	0.975quant
SD for the Gaussian observations	6.24889	6.14644	6.35186
SD for the Gaussian observations1	6.49417	6.39819	6.55571
SD for SES	1.84044	1.38494	2.53273
SD for SES1	0.39571	0.07779	1.24123

sd prior, post



observations τ



ses effect σ

RW1 v RW2

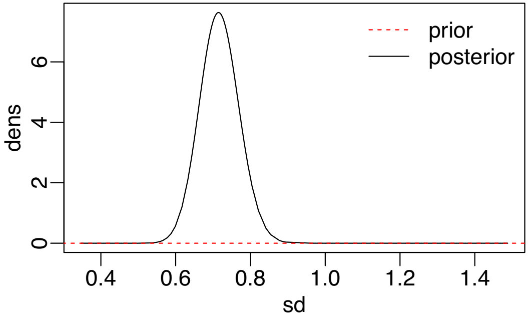
- The difference is what happens when $\sigma = 0$
- RW1, $U_i = U_0$
- RW2, $U_i = U_0 + i \cdot (U_1 - U_0)$
- RW2 doesn't need to work to make straight lines
- RW1 needs large σ to capture SES effect
- Use RW2 if you regard linear regression as the 'default' or 'null'

An inappropriate model for deaths

```r
> deathsBayesUnwise = inla(
+   Value ~ month +
+    f(timeRw, model='rw2',
+      hyper = list(theta=list(
+        prior='pc.prec', scale.model=FALSE,
+        param=c(0.02, 0.05)))  +
+     offset(nDays),
+   data=newData,
+   family='poisson',
+   control.predictor = list(compute = TRUE, link=1)
+ )
```

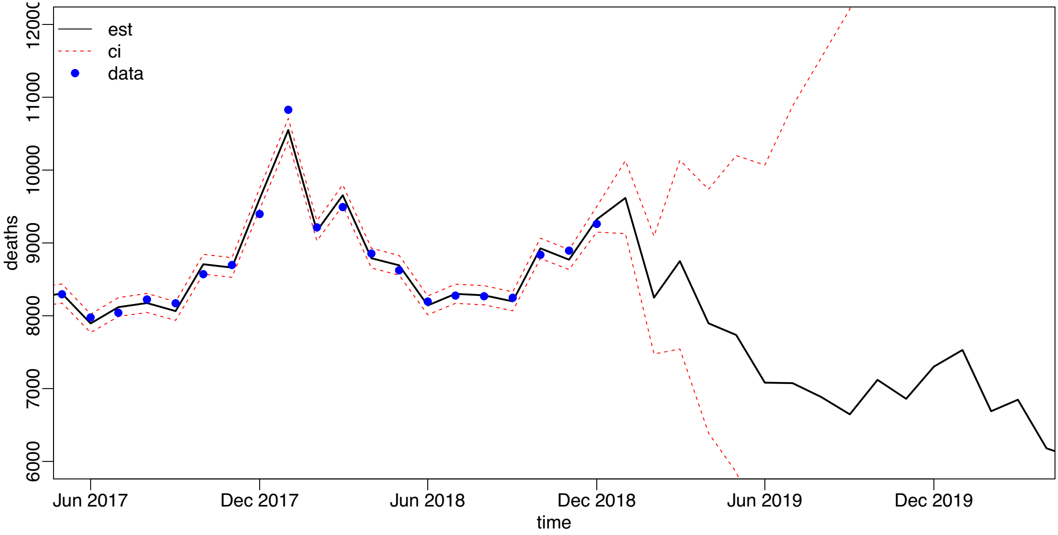
- Eliminating the iid random effect
- Any variation in death counts about the mean is due to the Poisson
- Not consistent with flu outbreaks

prior, post



- The time effect will try to capture all extra-Poisson variation
- so the SD is much larger
- larger even than the iid SD was
- If we believe this model, deaths are very unstable over time

Predicted values



Constraints and identifiability

- Suppose $U_{t+1} \sim \mathsf{N}(U_t, \sigma^2)$, an RW1
- and $\eta_t = \mu + U_t$
- not identifiable
- $\mu_2 = \mu + 10000$ and $V_t = U_t - 10000$
- $\gamma_t = \mu_2 + V_t$ has the same distribution as η_t
- ... and V_t is also an RW1
- must *constrain* U_t for the model to be identifiable
- default in `inla` is $\sum_t U_t = 0$
- interpret μ as an average η_t
- alternative is to set $U_k = 0$ for some k
- then $\mu = \eta_k$

Ontario deaths with constraint

a bit complicated, so not shown

Generalized Additive Mixed Models

$$
\begin{aligned}
Y_{ij} &\sim (\lambda_{ij}, \theta) \\
g(\lambda_{ij}) &= X_{ij}\beta + f(W_{ij}) + U_i \\
[U_1 \ldots U_M]^\mathsf{T} &\sim \mathsf{MVN}(0, \Sigma)
\end{aligned}
$$

- Random effects are possible in a semi-parametric model
- School level random effect should be in the math model
- with Bayes/INLA this is conceptually and computationally straightforward
- These models are possible, but far from rigorous, using penalized likelihood with cross validation.
- the `gamm` function alternates between `gam` and `glmmPQL`

INLA v GAM

...or more accurately, joint prob + xv versus Bayes

GAM's

- fast and seem less likely to break
- give the appearance of making few assumptions

Bayes

- fast enough using INLA (or BayesX)
- explicit specification of assumptions
- rigorous inference methodology

Bayes v GAM

Anti-GAM

- Uncertainty and inference with GAM's much be approached with caution
- GAM's assumptions are implicit
- ...and are quite strong
- so GAM's probably don't always adequately reflect uncertainty in their estimates

Pro-GAM

- Stability of GAM's make them a better sludgehammer than Bayes
- ...for prediction problems with Big Data
- Brieman would approve

Arguments for semi-parametric Bayes

- Forces one to be specific about the model
- Cox would prefer Bayes to GAM, provided you approach it properly
  - Think carefully about the model before you start, because an overly complex model will break
  - Use scientific knowledge to decide where to go non-parametric, before looking at the data
  - If you use models which the data don't support, your results should be crazy!
- Bayes is more extensible
  - Spatial correlations
  - Censored survival times

References I

- Brown, Patrick E and P de Jong (2001). "Nonparametric smoothing using state space techniques". In: *Canadian Journal of Statistics* 29.1, pp. 37–50. DOI: 10.2307/3316049.
- Eilers, Paul HC and Brian D Marx (1996). "Flexible smoothing with B-splines and penalties". In: *Statistical science*, pp. 89–102.
- Green, Peter J and Bernard W Silverman (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.
- Hastie, Trevor J and Robert J Tibshirani (1990). *Generalized additive models*. Vol. 43. CRC Press.
- Heckman, Nancy E. and James O. Ramsay (2000). "Penalized Regression with Model-Based Penalties". In: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 28.2, pp. 241–258. URL: http://www.jstor.org/stable/3315976.

References II

Myllymäki, Mari, Tomáš Mrkvička, Pavel Grabarnik, Henri Seijo, and Ute Hahn (2017). "Global envelope tests for spatial processes". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.2, pp. 381–404. DOI: 10.1111/rssb.12172.

Simpson, Daniel, Haavard Rue, Andrea Riebler, Thiago G. Martins, and Sigrunn H. Sorbye (Feb. 2017). "Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors". In: *Statistical Science* 32.1, pp. 1–28. DOI: 10.1214/16-STS576.

Wood, Simon (2006). *Generalized additive models: an introduction with R*. CRC press.