Essex Summer School in Social Science Data Analysis
Panel Data Analysis for Comparative Research

Modeling Stationary Time Series

Christopher Adolph

Department of Political Science
and
Center for Statistics and the Social Sciences
University of Washington, Seattle

The story so far

We've learned:

- how to decide whether one estimator is "better" than another under a given DGP

- why our LS models don't work well with time series

- how to obtain quantities of interest, such as $\mathbb{E}(y|x_c)$ from an estimated model

- the basics of time series dynamics, including:

trends, autoregression, moving averages, seasonality, stationarity

What we're doing today

Next steps:

- Use ML to estimate AR(p), MA(q), and ARMA(p,q) models for stationary series
- Use our time series knowledge & MLE fitting tools to select p and q
- Use simulations to understand how $\mathbb{E}(y_t|x_t)$ changes as we vary x_t over time

An AR(1) Regression Model

To create a regression model for an AR(1) process,
we allow the mean of the process to shift by adding c_t to the equation:

$$y_t = y_{t-1}\phi_1 + c_t + \varepsilon_t$$

We then parameterize c_t as the sum of a set of time varying covariates,

$x_{1t}, x_{2t}, x_{3t}, \ldots$

and their associated parameters,

$\beta_1, \beta_2, \beta_3, \ldots$

which we compactly write in matrix notation as $c_t = \mathbf{x}_t\boldsymbol{\beta}$

An AR(1) Regression Model

Substituting for c_t, we obtain the AR(1) regression model:

$$y_t = y_{t-1}\phi_1 + \mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t$$

Estimation is by maximum likelihood, *not* LS

(We will discuss the LS version later)

MLE accounts for dependence of y_t on past values; complex derivation (see James Hamilton, *Time Series Analysis* for a review)

We'll focus on interpreting this model in practice

Aside: the AR(1) likelihood function

Why is the MLE for AR(1) more complex than the MLE for linear regression?

Suppose our time series "starts" at $t = 1$:
there is no lag before $t = 1$, so period 1 has no AR(1) term

Then the distribution of the first observation is

$$y_1 \sim \mathcal{N}(\mathbf{x}_1 \boldsymbol{\beta}, \sigma^2)$$

But after $t = 1$, y_t is AR(1), so y_{t+1} depends on y_t

$$\begin{aligned} y_2 | y_1 &\sim \mathcal{N}(\mathbf{x}_2 \boldsymbol{\beta} + \phi y_1, \sigma^2) \\ y_3 | y_2 &\sim \mathcal{N}(\mathbf{x}_3 \boldsymbol{\beta} + \phi y_2, \sigma^2) \end{aligned}$$

and so on up to the distribution of y_t

This means the y_t's are not iid: the usual Normal MLE is inadequate,
and we must create a new likelihood based on the distributions above

Aside: the AR(1) likelihood function

Multiplying together the pdfs of the distributions of $y_1, \ldots y_t$
and reducing to sufficient statistics yields the following log-likelihood for AR(1):

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \phi_1 | \mathbf{y}, \mathbf{X}) &= -\frac{1}{2}\log\left(\frac{\sigma^2}{1-\phi_1^2}\right) - \frac{\left(y_1 - \frac{\mathbf{x}_1\boldsymbol{\beta}}{1-\phi_1}\right)^2}{\frac{2\sigma^2}{1-\phi_1^2}} \\ & \quad -\frac{T-1}{2}\log\sigma^2 - \sum_{t=2}^{T}\frac{(y_t - \mathbf{x}_t\boldsymbol{\beta} - \phi_1 y_{t-1})^2}{2\sigma^2}\end{aligned}$$

Only differs from least squares in the treatment of y_1,
so very similar to OLS with a lagged DV if T is large

But LS standard errors can be substantially biased if T is small

The definition of “small” depends on ϕ, σ, and covariates,
so you try both the AR(1) MLE and OLS if you are worried!

Aside: the AR(1) likelihood function

Multiplying together the pdfs of the distributions of $y_1, \ldots y_t$
and reducing to sufficient statistics yields the following log-likelihood for AR(1):

$$\mathcal{L}(\boldsymbol{\beta}, \phi_1 | \mathbf{y}, \mathbf{X}) = -\frac{1}{2}\log\left(\frac{\sigma^2}{1-\phi_1^2}\right) - \frac{\left(y_1 - \frac{\mathbf{x}_1\boldsymbol{\beta}}{1-\phi_1}\right)^2}{\frac{2\sigma^2}{1-\phi_1^2}}$$

$$-\frac{T-1}{2}\log\sigma^2 - \sum_{t=2}^{T}\frac{(y_t - \mathbf{x}_t\boldsymbol{\beta} - \phi_1 y_{t-1})^2}{2\sigma^2}$$

MLEs only get more complex as we move towards ARMA(p,q)

Generally, we can treat ARMA estimation as a black box

Our main concern will be how to select the right model
and interpret what it means substantively

Interpreting AR(1) parameters

Suppose that a country's GDP follows this simple model

$$\begin{aligned}\mathrm{GDP}_t &= \phi_1 \mathrm{GDP}_{t-1} + \beta_0 + \beta_1 \mathrm{Democracy}_t + \varepsilon_t \\ \mathrm{GDP}_t &= 0.9 \times \mathrm{GDP}_{t-1} + 10 + 2 \times \mathrm{Democracy}_t + \varepsilon_t\end{aligned}$$

Suppose that at year t, $\mathrm{GDP}_t = 100$,
and the country is a non-democracy, $\mathrm{Democracy}_t = 0$.

What would happen if we "made" this country a democracy in period $t + 1$?

Interpreting AR(1) parameters

$$y_t = y_{t-1}\phi_1 + \mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t$$

Recall:
an AR(1) process can be viewed as the geometrically declining sum of all its past errors.

When we add the time-varying mean $x_t\boldsymbol{\beta}$ to the equation, the following now holds:

$$y_t = (\mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t) + \phi_1(\mathbf{x}_{t-1}\boldsymbol{\beta} + \varepsilon_{t-1}) + \phi_1^2(\mathbf{x}_{t-2}\boldsymbol{\beta} + \varepsilon_{t-2}) + \phi_1^3(\mathbf{x}_{t-3}\boldsymbol{\beta} + \varepsilon_{t-3}) + \ldots$$

That is, y_t represents the sum of all past x_t's as filtered through $\boldsymbol{\beta}$ and ϕ_1

Interpreting AR(1) parameters

Take a step back:
suppose c_t is actually fixed for all time at c,
so that $c = c_t$

Now, we have

$$\begin{aligned} y_t &= (c+\varepsilon_t) + \phi_1(c+\varepsilon_{t-1}) + \phi_1^2(c+\varepsilon_{t-2}) + \phi_1^3(c+\varepsilon_{t-3}) + \dots \\ &= \frac{c}{1-\phi_1} + \varepsilon_t + \phi_1\varepsilon_{t-1} + \phi_1^2\varepsilon_{t-2} + \phi_1^3\varepsilon_{t-3}\dots \end{aligned}$$

which follows from the limits for infinite series

Taking expectations removes everything but the first term:

$$\mathbb{E}(y_t) = \frac{c}{1-\phi_1}$$

Implication:
if, starting at time t and going forward to ∞,
we fix $\mathrm{x}_t\boldsymbol{\beta}$,
then y_t will converge to $\mathrm{x}_t\boldsymbol{\beta}/(1-\phi_1)$

Interpreting AR(1) parameters

$$\begin{aligned}
\mathrm{GDP}_t &= \phi_1 \mathrm{GDP}_{t-1} + \beta_0 + \beta_1 \mathrm{Democracy}_t + \varepsilon_t \\
\mathrm{GDP}_t &= 0.9 \times \mathrm{GDP}_{t-1} + 10 + 2 \times \mathrm{Democracy}_t + \varepsilon_t
\end{aligned}$$

If at year t, $\mathrm{GDP}_t = 100$ and the country is a non-democracy ($\mathrm{Democracy}_t = 0$) then:

This country is in a steady state –
it will tend to have GDP of 100 every period, with small errors from ε_t (verify this)

Interpreting AR(1) parameters

$$\begin{aligned}\mathrm{GDP}_t &= \phi_1 \mathrm{GDP}_{t-1} + \beta_0 + \beta_1 \mathrm{Democracy}_t + \varepsilon_t \\ \mathrm{GDP}_t &= 0.9 \times \mathrm{GDP}_{t-1} + 10 + 2 \times \mathrm{Democracy}_t + \varepsilon_t\end{aligned}$$

Now suppose we make the country a democracy in period $t+1$,
so that $\mathrm{Democracy}_{t+1} = 1$.

The model predicts that in period $t+1$, the level of GDP will rise by $\beta = 2$, to 102.

This *appears* to be a small effect, but. . .

Interpreting AR(1) parameters

$$\begin{aligned}\mathrm{GDP}_t &= \phi_1 \mathrm{GDP}_{t-1} + \beta_0 + \beta_1 \mathrm{Democracy}_t + \varepsilon_t \\ \mathrm{GDP}_t &= 0.9 \times \mathrm{GDP}_{t-1} + 10 + 2 \times \mathrm{Democracy}_t + \varepsilon_t\end{aligned}$$

. . . the effect accumulates, so long as $\mathrm{Democracy} = 1$

$\mathbb{E}(\hat{y}_{t+2}|x_{t+2}) = 0.9 \times 102 + 10 + 2 = 103.8$

$\mathbb{E}(\hat{y}_{t+3}|x_{t+3}) = 0.9 \times 103.8 + 10 + 2 = 105.42$

$\mathbb{E}(\hat{y}_{t+4}|x_{t+4}) = 0.9 \times 105.42 + 10 + 2 = 106.878$

. . .

$\mathbb{E}(\hat{y}_{t=\infty}|x_{t=\infty}) = (10 + 2)/(1 - 0.9) = 120$

So is this a big effect or a small effect?

Interpreting AR(1) parameters

$\mathbb{E}(\hat{y}_{t=\infty}|x_{t=\infty}) = (10+2)/(1-0.9) = 120$

So is this a big effect or a small effect?

It depends on the length of time your covariates remain fixed.

Many social variables change rarely, so their effects accumulate slowly over time (e.g., institutions)

Presenting only β_1, rather than the accumulated change in y_t after x_t changes, could drastically *understate* the relative substantive importance of our social & political covariates compared to rapidly changing covariates

This understatement gets larger the closer ϕ_1 gets to 1
—which is where our ϕ_1's tend to be!

A catch: remember that if $\phi_1 = 1$, long-run predictions are impossible,
so forecasting will produce misleading results of nonstationary processes

Interpreting AR(1) parameters

Recommendation:
Simulate the change in y_t given a change in x_t through enough periods to capture the real-world impact of your variables

If you are studying partisan effects, and new parties tend to stay in power 5 years, don't report β_1 or the one-year change in y. Iterate out to five years.

What is the confidence interval around these cumulative changes in y given a permanent change in x?

A complex function of the se's of $\boldsymbol{\phi}$ and $\boldsymbol{\beta}$

So simulate out to y_{t+k} using draws from the estimated distributions of $\hat{\phi}$ and $\hat{\boldsymbol{\beta}}$

`R` will help with this, using `predict()` and (in `simcf`), `ldvsimev()`

Example: UK vehicle accident deaths

Number of monthly deaths and serious injuries in UK road accidents

Data range from January 1969 to December 1984.

In February 1983, a new law requiring seat belt use took effect

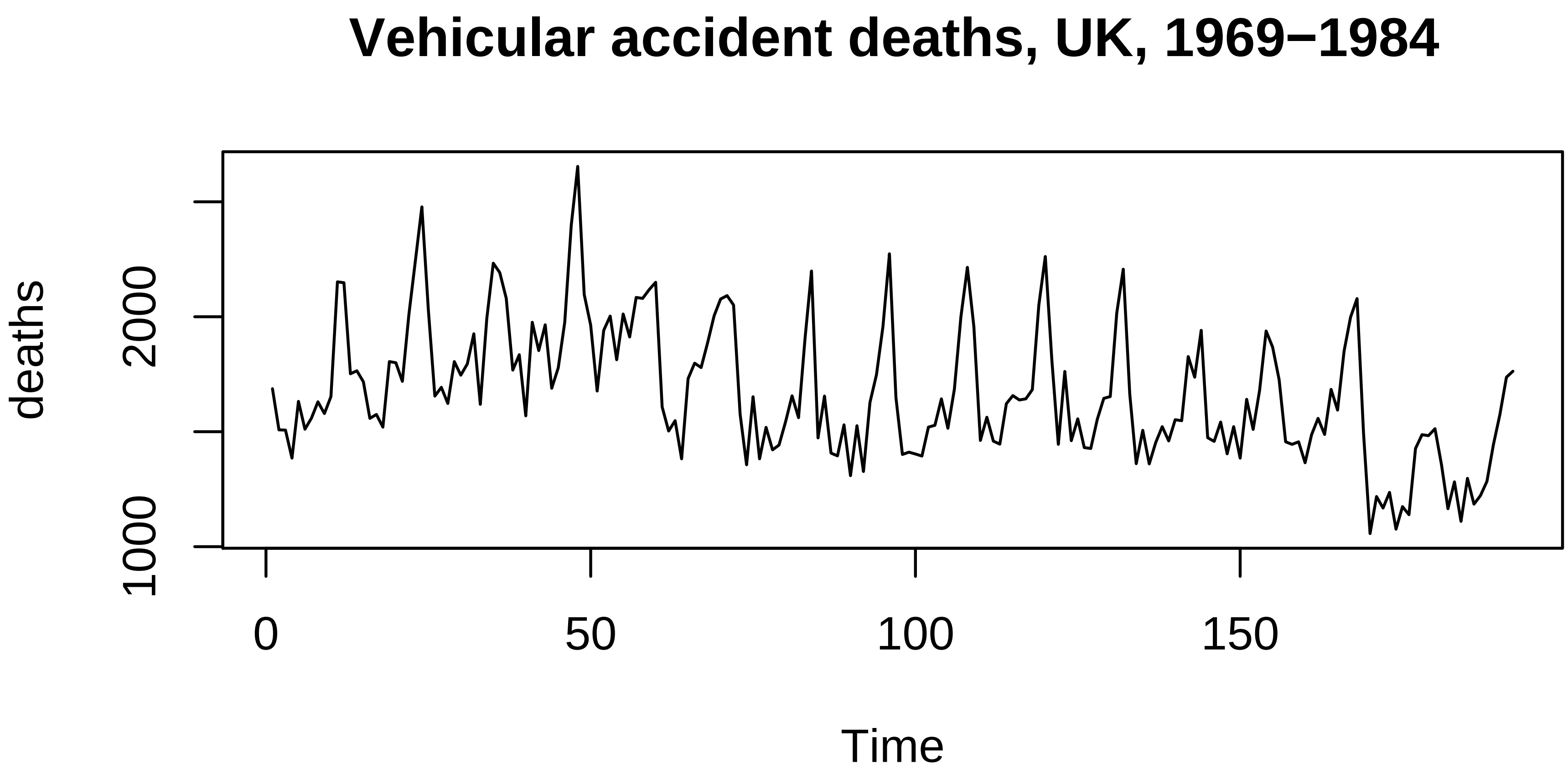
Source: Harvey, 1989, p.519ff.

`http://www.staff.city.ac.uk/~sc397/courses/3ts/datasets.html`

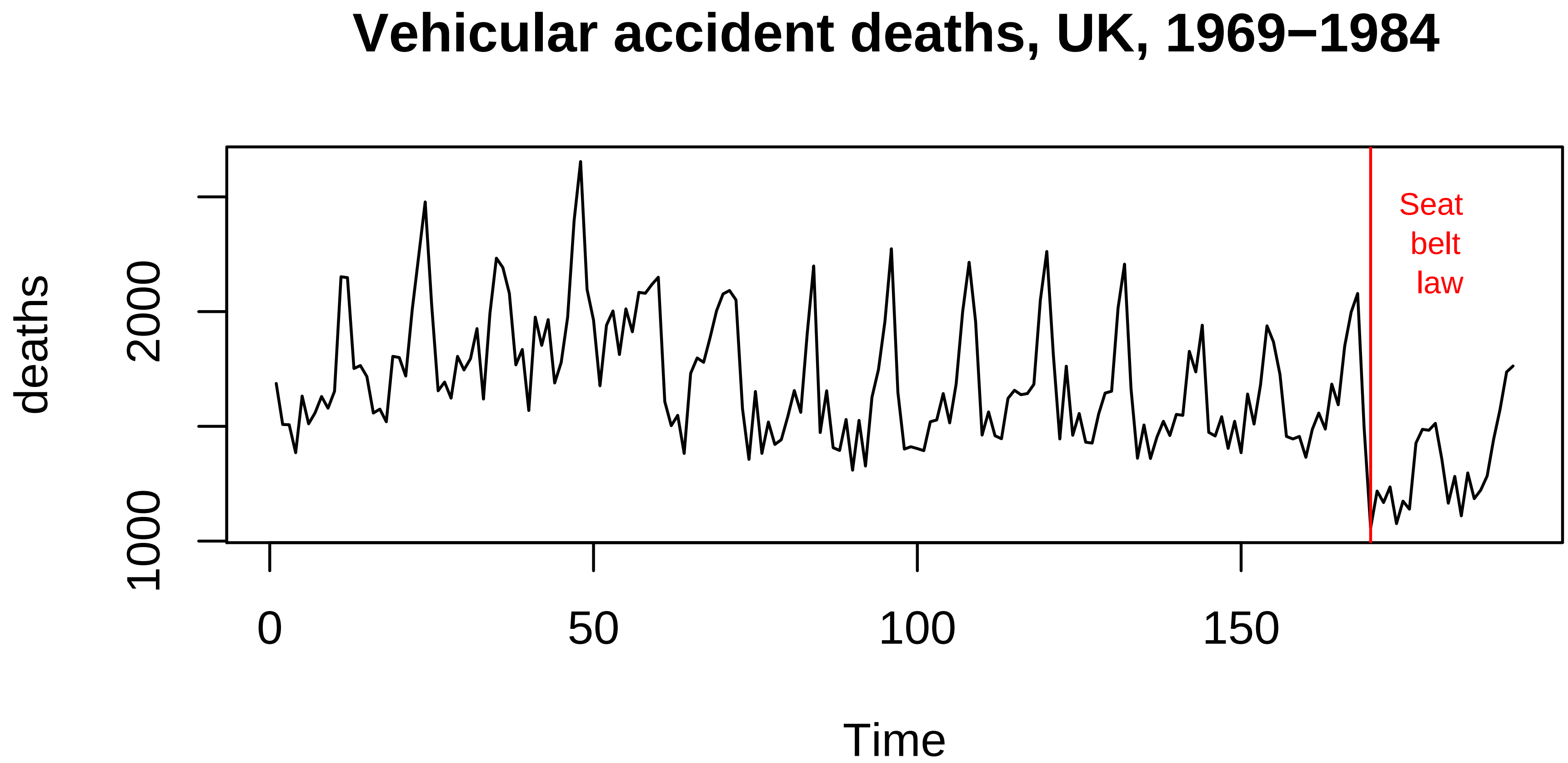
Simple, likely stationary data

Possibly seasonal

Simplest possible covariate: a single dummy



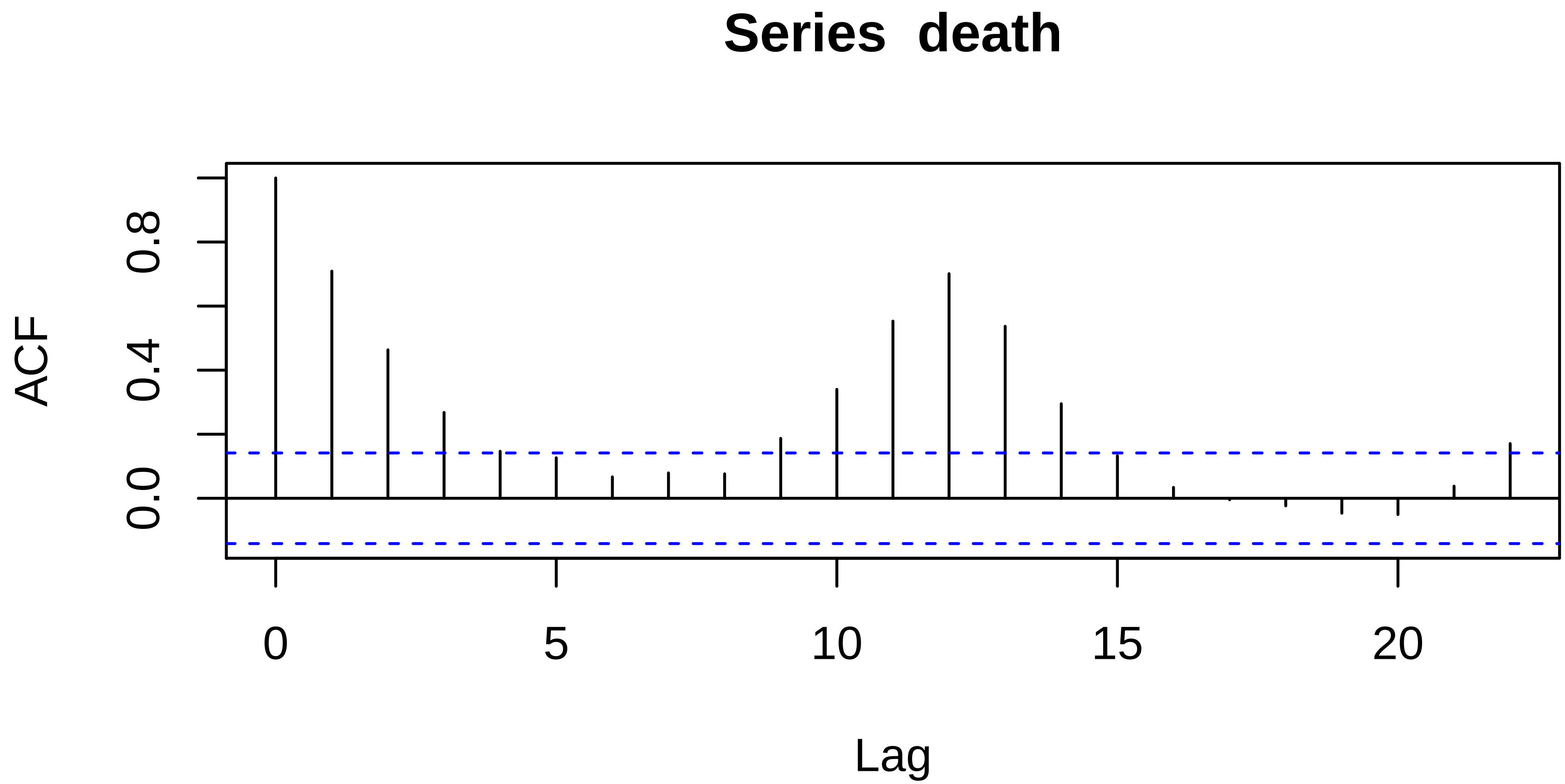
The time series itself – looks cyclical, with a break in the series



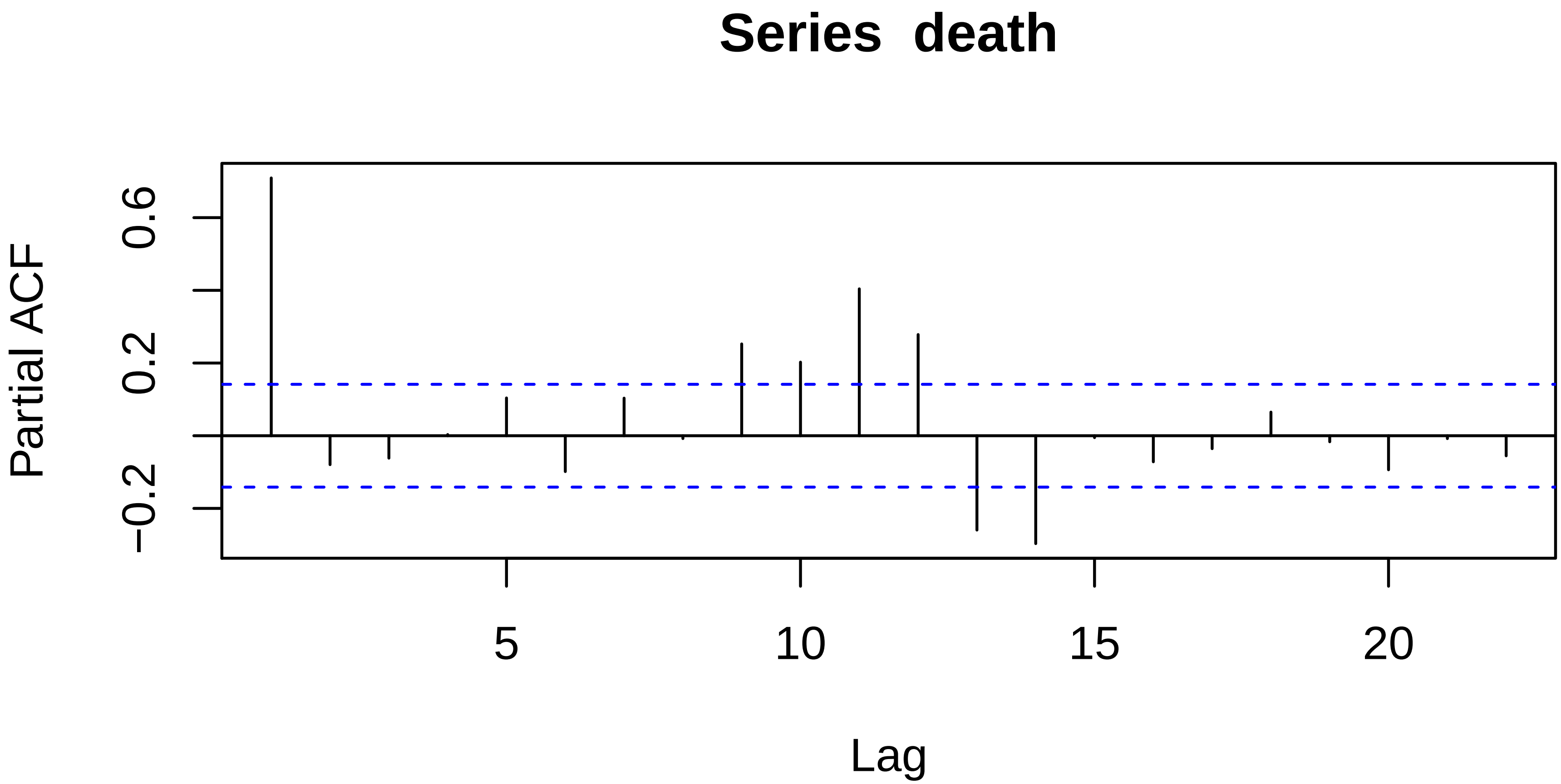
The break corresponds closely with the change in seat belt laws

In a real data analysis, everything past this point is a bit gratuitous–
this time series plot is simple and persuasive

But as most data analyses are more complex,
this is a good testbed to learn techniques

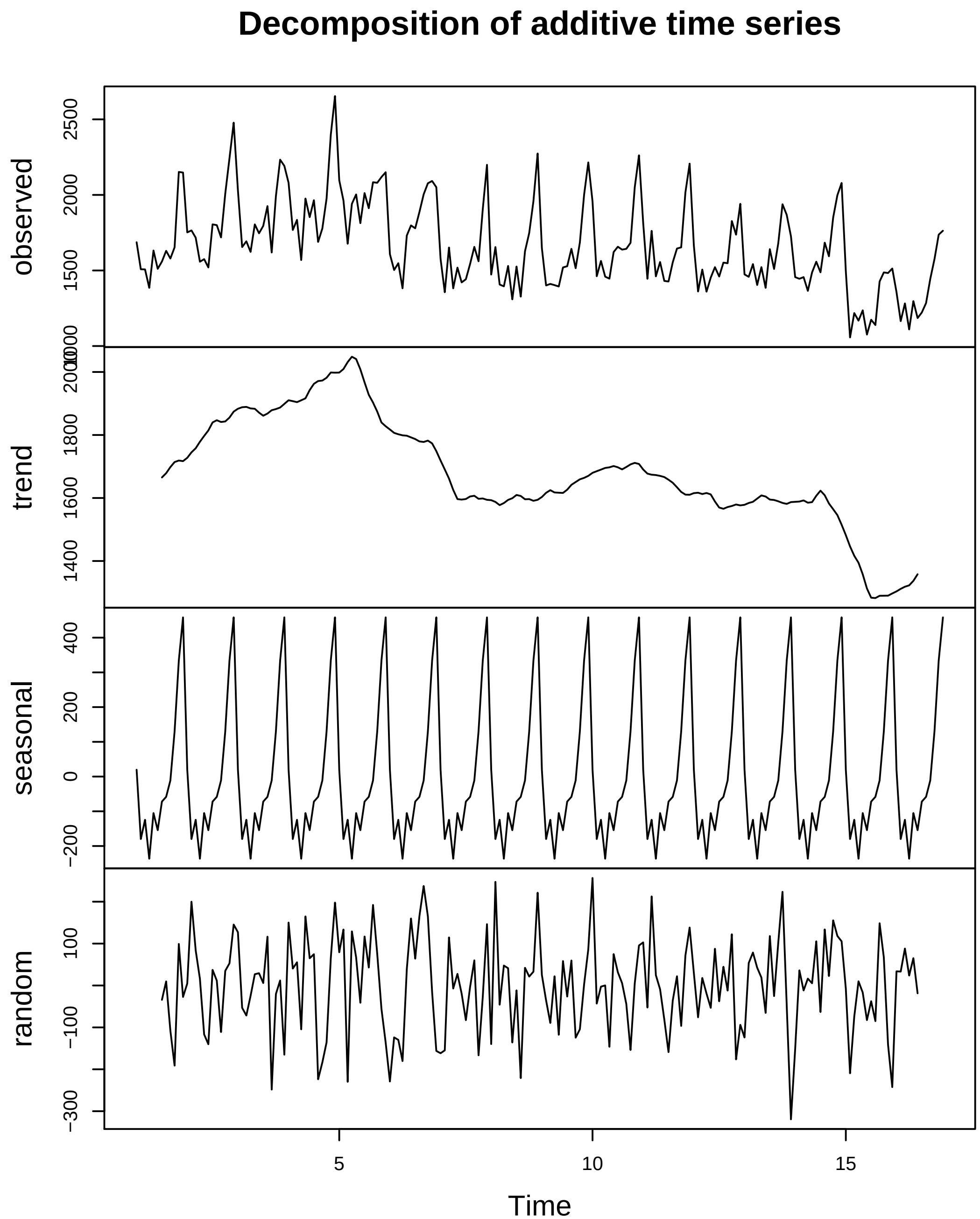


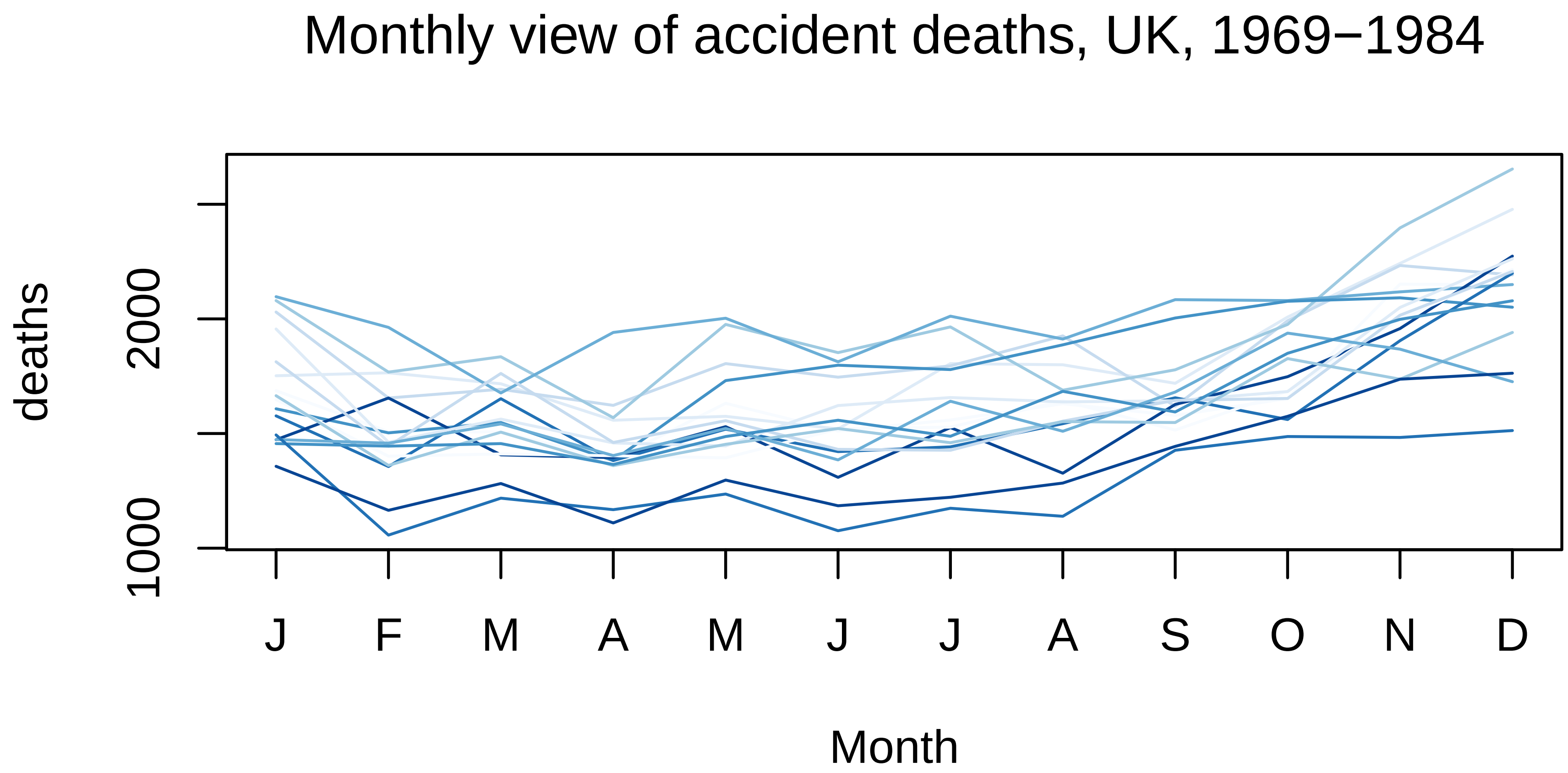
What does this suggest?



What does this suggest?

How should we model seasons?


Decomposition of additive time series
observed
trend
seasonal
random
Time
1000
1500
2000
2500
1400
1600
1800
2000
−200
0
200
400
−300
−100
100
5
10
15



November and December look especially dangerous

October and January look a bit dangerous

We could control for each month, select months, or Q4

This might also depend on serial correlation

Model 1a: AR(1) specification

```
## Estimate an AR(1) using arima
xcovariates <- law
arima.res1a <- arima(death, order = c(1,0,0),
                     xreg = xcovariates, include.mean = TRUE
                     )
```

```
Coefficients:
        ar1  intercept  xcovariates
      0.644    1719.19       -377.5
s.e.  0.055      42.08        107.7

sigma^2 estimated as 39289:  log likelihood = -1288,  aic = 2585
```

We begin with a simple model ignoring seasonality,
and controlling for one autoregressive lag

Model 1b: AR(1) specification with Q4 control

```
## Estimate an AR(1) using arima
xcovariates <- cbind(law, q4)
arima.res1b <- arima(death, order = c(1,0,0),
                     xreg = xcovariates, include.mean = TRUE
                     )
```

```
Coefficients:
         ar1  intercept        law        q4
      0.5352  1638.0301  -395.6701  324.5653
s.e.  0.0636    28.1199    72.3030   34.5033

sigma^2 estimated as 26669:  log likelihood = -1250.97,  aic = 2511.93
```

Model 1c: AR(1) specification with all months

```
## Estimate an AR(1) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res1c <- arima(death, order = c(1,0,0),
                     xreg = xcovariates, include.mean = TRUE
                     )
```

```
Coefficients:
         ar1  intercept        law      jan       feb
      0.6442  1638.6270  -370.0694  81.3021  -95.1350
s.e.  0.0550    42.9093    70.2727  54.8127   54.5036
          mar        apr       may       jun      aug
     -44.3298  -157.3445  -19.9428  -75.6674  14.7670
s.e.  53.0792    50.2149   45.0247   35.1890  35.1882
         sep       oct       nov      dec
      67.4890  206.6686  405.9134  522.0696
s.e.  45.0184   50.1913   53.0074   54.3054

sigma^2 estimated as 16333:  log likelihood = -1204,  aic = 2437.99
```

Model 1d: AR(1) specification with select months

```
## Estimate an AR(1) using arima
xcovariates <- cbind(law, jan, sep, oct, nov, dec)
arima.res1d <- arima(death, order = c(1,0,0),
                     xreg = xcovariates, include.mean = TRUE
                     )

Coefficients:
         ar1  intercept        law       jan      sep
      0.6045  1589.4405  -377.7457  154.7288  80.7422
s.e.  0.0575    29.4161    69.7719   35.7336  35.8534
           oct        nov       dec
      238.3880   451.3567  579.9770
s.e.   42.6836    44.3474   42.6108

sigma^2 estimated as 18989:  log likelihood = -1218.42,  aic = 2454.83
```

Model 1e: AR(1)AR(1)$_{12}$ specification

```
## Estimate an AR(1)AR(1)_12 using arima
xcovariates <- cbind(law)
arima.res1e <- arima(death, order = c(1,0,0),
                     seasonal = list(order = c(1,0,0), period = 12),
                     xreg = xcovariates, include.mean = TRUE
                     )
```

```
Coefficients:
         ar1    sar1  intercept        law
      0.4446  0.6511  1710.1531  -347.6812
s.e.  0.0695  0.0564    53.3648    73.0634

sigma^2 estimated as 23693:  log likelihood = -1242.86,  aic = 2495.71
```

Model 1e: AR(1)AR(1)$_{12}$ specification

Two questions:

1. Which model to select?

Additive or multiplicative seasonality?

A full set of month dummies, or a selection?

2. What is the effect of adding the law?

In period $t+1$? $\quad t+12$? $\quad t+60$

How "significant" is this effect over those periods?

Summary of fit so far

Model	Components	AIC	$\hat{\beta}_{\text{Law}}$	$\text{se}(\hat{\beta}_{\text{Law}})$
1a	AR(1)	2585	-377	108
1b	AR(1), q4	2512	-396	72
1c	AR(1), all months	2438	-370	70
1d	AR(1), sep to jan	2455	-378	70
1e	AR(1)AR(1)$_{12}$	2496	-348	73

Which is the best fitting approach to seasonality?

Why did I use AIC to select models? What might be better?

What substantive difference does it make?

And what about higher order serial correlation?

An AR(p) Regression Model

The AR(p) regression model is a straightforward extension of the AR(1)

$$y_t = y_{t-1}\phi_1 + y_{t-2}\phi_2 + \ldots + y_{t-p}\phi_p + \mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t$$

Estimation is again by MLE, but similar to OLS with p lags of DV if t is large; MLE differs only in treatment of y_1 to y_p

Note that for fixed mean, y_t now converges to

$$\mathbb{E}(y_t) = \frac{c}{1 - \phi_1 - \phi_2 - \phi_3 - \ldots - \phi_p}$$

Implication:
if, starting at time t and going forward to ∞, we fix $\mathrm{x}_i\boldsymbol{\beta}$,
then y_t will converge to $\mathrm{x}_i\boldsymbol{\beta}/(1 - \phi_1 - \phi_2 - \phi_3 - \ldots - \phi_p)$

Estimation and interpretation similar to above & uses same `R` functions

MA(1) Models

To create a regression model for an MA(1) process:

$$y_t = \varepsilon_{t-1}\rho_1 + \mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t$$

Estimation is again by maximum likelihood;
no there is no obvious approximation to least squares

Once again a complex procedure, but still a generalization of the Normal MLE

Any dynamic effects in this model are quickly mean reverting

ARMA(p,q): Putting it all together

To create a regression model for an ARMA(p,q) process:

$$y_t = y_{t-1}\phi_1 + y_{t-2}\phi_2 + \ldots + y_{t-p}\phi_p + \varepsilon_{t-1}\rho_1 + \varepsilon_{t-2}\rho_2 + \ldots + \varepsilon_{t-q}\rho_q + \mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t$$

We will need an MLE to obtain $\hat{\boldsymbol{\phi}}$, $\hat{\boldsymbol{\rho}}$, and $\hat{\boldsymbol{\beta}}$

Once again a complex procedure, but still a generalization of the Normal case

Note the AR(p) process dominates in two senses:

- Stationarity determined just by AR(p) part of ARMA(p,q)
- Long-run level determined just by AR(p) terms: still $\mathbf{x}_i\boldsymbol{\beta}/(1 - \sum_p \phi_p)$

Model 2a: AR(2) specification

```
## Estimate an AR(2) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res2a <- arima(death, order = c(2,0,0),
                     xreg = xcovariates, include.mean = TRUE
                     )


Coefficients:
         ar1     ar2  intercept        law      jan
      0.4696  0.2711  1635.0869  -347.9213  83.7469
s.e.  0.0692  0.0694    45.6076    80.5683  46.9299
          feb       mar        apr       may       jun
      -94.9882  -44.0442  -157.2316  -19.8376  -75.5957
s.e.   46.5145   45.0452    42.8448   37.9719   35.0631
         aug      sep       oct       nov       dec
      14.8059  67.5047  206.7362  406.0569  522.4596
s.e.  35.0623  37.9640   42.8242   44.9760   46.4368

sigma^2 estimated as 15118:  log likelihood = -1196.65,  aic = 2425.3
```

Model 2b: MA(1) specification

```
## Estimate an MA(1) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res2b <- arima(death, order = c(0,0,1),
                     xreg = xcovariates, include.mean = TRUE
                     )


Coefficients:
         ma1  intercept        law      jan       feb
      0.4539  1641.4834  -391.7280  79.9732  -94.6320
s.e.  0.0538    39.7814    45.5288  55.5797   55.6807
          mar        apr       may       jun      aug
      -44.0097  -157.2155  -19.8754  -75.6604  14.8400
s.e.   55.6807    55.6807   55.6807   43.9719  43.9719
          sep       oct       nov       dec
      67.6897  207.0297  406.5988  522.4457
s.e.  55.6807   55.6807   55.6807   55.5411

sigma^2 estimated as 20566:  log likelihood = -1225.97,  aic = 2481.93
```

Model 2c: ARMA(1,1) specification

```
## Estimate an ARMA(1,1) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res2c <- arima(death, order = c(1,0,1),
                     xreg = xcovariates, include.mean = TRUE
                     )


Coefficients:
         ar1      ma1  intercept        law      jan
      0.9349  -0.5994  1629.5549  -323.4929  85.7471
s.e.  0.0383   0.1076    58.6795    83.2081  40.4544
          feb       mar        apr       may       jun
      -94.0923  -43.6000  -156.8606  -19.6467  -75.5028
s.e.   40.2349   39.7247    38.8954   37.7225   36.1673
         aug      sep       oct       nov       dec
      14.7339  67.3872  206.5916  405.9572  522.3735
s.e.  36.1671  37.7207   38.8896   39.7111   40.2083

sigma^2 estimated as 14568:  log likelihood = -1193.18,  aic = 2418.37
```

Model 2d: ARMA(2,1) specification

```
## Estimate an ARMA(2,1) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res2d <- arima(death, order = c(2,0,1),
                     xreg = xcovariates, include.mean = TRUE)
Coefficients:
         ar1      ar2      ma1  intercept        law
      1.1899  -0.2157  -0.7950  1626.1862  -321.2201
s.e.  0.1071   0.0976   0.0724    68.6982    78.8301
          jan       feb       mar        apr       may
      84.8843  -94.5311  -43.8782  -157.0544  -19.7871
s.e.  41.3869   41.3010   41.0435    40.5352   39.3222
          jun      aug      sep       oct       nov
     -75.5646  14.8208  67.5749  206.8634  406.3691
s.e.  35.1484  35.1483  39.3216   40.5327   41.0341
          dec
     522.9159
s.e.  41.2487

sigma^2 estimated as 14284:  log likelihood = -1191.33,  aic = 2416.66
```

Model 2e: ARMA(1,2) specification

```
## Estimate an ARMA(1,2) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res2e <- arima(death, order = c(1,0,2),
                     xreg = xcovariates, include.mean = TRUE)
Coefficients:
         ar1      ma1      ma2  intercept       law
      0.9620  -0.5892  -0.1228   1627.146  -322.6854
s.e.  0.0253   0.0752   0.0705     66.814    79.2449
         jan       feb       mar        apr       may
     85.1562  -94.1511  -43.6591  -156.9126  -19.6915
s.e. 40.7504   40.6400   40.3701    39.9498   39.3736
          jun      aug      sep       oct       nov
     -75.5237  14.7645  67.4691  206.7084  406.1477
s.e.  35.5453  35.5453  39.3730   39.9476   40.3650
          dec
     522.6613
s.e.  40.5994

sigma^2 estimated as 14356:  log likelihood = -1191.82,  aic = 2417.63
```

Model 2f: ARMA(2,2) specification

```
## Estimate an ARMA(2,2) using arima
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct,
                     nov, dec)
arima.res2f <- arima(death, order = c(2,0,2),
                     xreg = xcovariates, include.mean = TRUE)
Coefficients:
         ar1     ar2     ma1      ma2  intercept
      0.0526  0.8449  0.3497  -0.6503  1625.7793
s.e.  0.0538  0.0413  0.1006   0.0998    61.5565
            law      jan       feb       mar       apr
      -312.2308  86.0931  -91.7482  -43.7677  -154.3960
s.e.    81.8335  40.9421   38.1258   40.4084    36.9053
           may       jun      aug      sep       oct
      -19.6984  -72.8430  17.6629  67.3856  209.8757
s.e.   38.9443   34.4385  34.4299  38.9431   36.8765
           nov       dec
      405.8869  526.1152
s.e.   40.3991   38.0647

sigma^2 estimated as 13794:  log likelihood = -1189.2,  aic = 2414.39
```

Whew!

This gets tedious fast. . .

To have R search automatically for a low AIC model,
try `auto.arima()` in the `forecast` library.

This gets complicated if the series is potentially seasonal and/or nonstationary

My practice: search/diagnose manually where feasible,
automatically where many runs are needed (e.g., 1 million time series analyses?)

More on this in lab. . .

Summary of fit

Model	Components	AIC	$\hat{\beta}_{\mathrm{Law}}$	$\mathrm{se}(\hat{\beta}_{\mathrm{Law}})$
1a	AR(1)	2585	-377	108
1b	AR(1), q4	2512	-396	72
1c	AR(1), all months	2438	-370	70
1d	AR(1), sep to jan	2455	-378	70
1e	AR(1)AR(1)$_{12}$	2496	-348	73
2a	AR(2), all months	2425	-348	81
2b	MA(1), all months	2482	-392	46
2c	ARMA(1,1), all months	2418	-323	83
2d,3a	ARMA(2,1), all months	2417	-321	79
2e	ARMA(1,2), all months	2418	-323	79
2f	ARMA(2,2), all months	2414	-321	79

Which model looks best?

What might be a better way to judge than AIC?

Cross-validation

Out of sample tests of fit are more reliable than in sample tests

But what is out-of-sample in time series?

Can't just pull random observations out of sequence:
best CV method for time series is a rolling forecast window

Issue for all cross-validation:
danger of collinearity if you have binary covariates that change rarely!

Summary of fit

Model	Components	AIC	cv1-MAE	β_{Law}	$\text{se}(\beta_{\text{Law}})$
1a	AR(1)	2585	120.4	-377	108
1b	AR(1), q4	2512	108.5	-396	72
1c	AR(1), all months	2438	83.9	-370	70
1d	AR(1), sep to jan	2455	119.7	-378	70
1e	AR(1)AR(1)$_{12}$	2496	119.7	-348	73
2a	AR(2), all months	2425	92.6	-348	81
2b	MA(1), all months	2482	79.9	-392	46
2c	ARMA(1,1), all months	2418	89.5	-323	83
2d,3a	ARMA(2,1), all months	2417	83.5	-321	79
2e	ARMA(1,2), all months	2418	84.8	-323	79
2f	ARMA(2,2), all months	2414	85.5	-321	79

We could look at the one-period-ahead out-of-sample forecast

cv1-MAE shows the mean absolute error in this prediction

Note we have fairly few periods left to forecast, as we need a long window to estimate the effect of the law (which doesn't start until period 170)

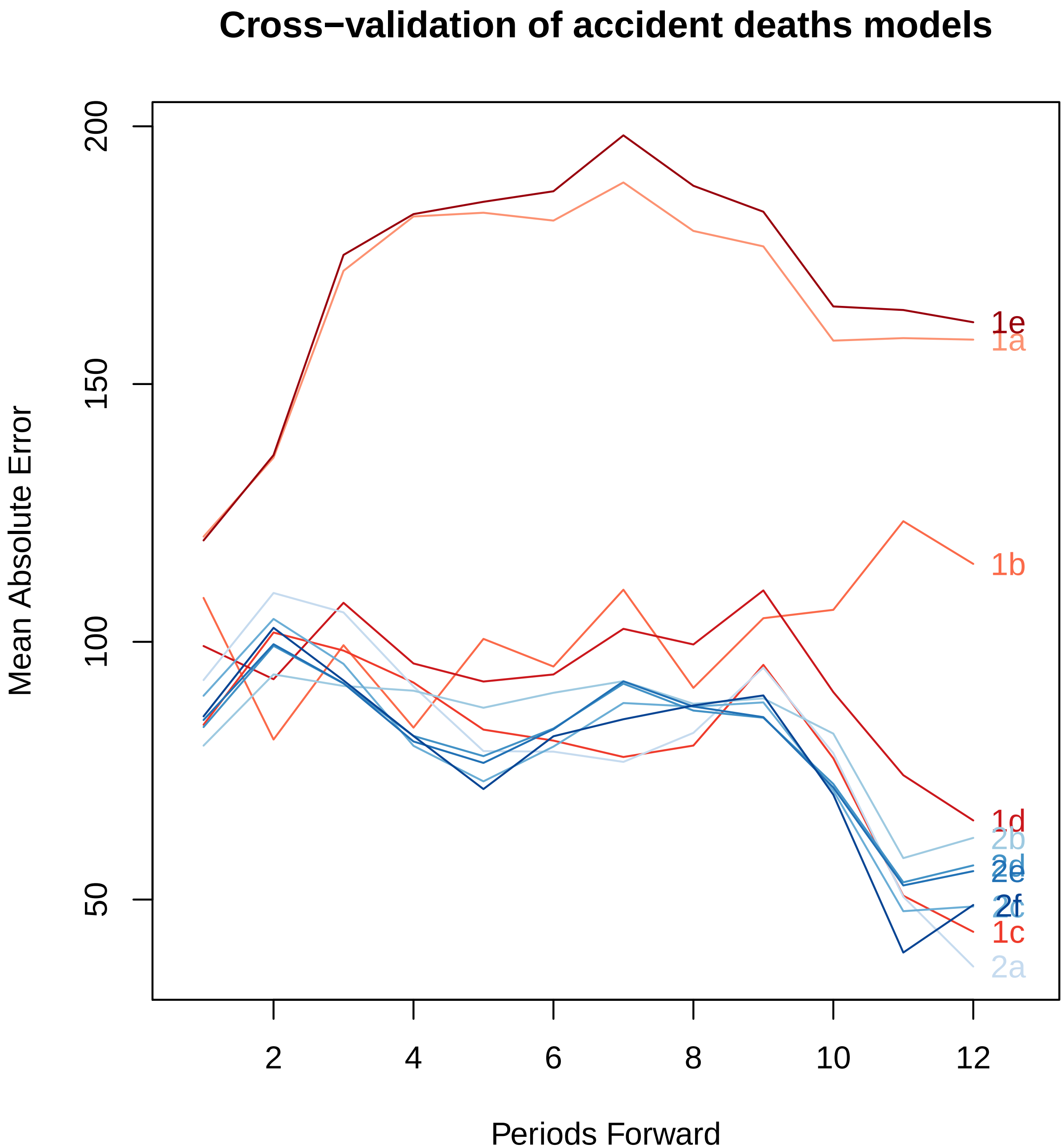
Summary of fit

Model	Components	AIC	cv1-MAE	β_{Law}	$\mathrm{se}(\beta_{\mathrm{Law}})$
1a	AR(1)	2585	120.4	−377	108
1b	AR(1), q4	2512	108.5	−396	72
1c	AR(1), all months	2438	83.9	−370	70
1d	AR(1), sep to jan	2455	119.7	−378	70
1e	AR(1)AR(1)$_{12}$	2496	119.7	−348	73
2a	AR(2), all months	2425	92.6	−348	81
2b	MA(1), all months	2482	79.9	−392	46
2c	ARMA(1,1), all months	2418	89.5	−323	83
2d,3a	ARMA(2,1), all months	2417	83.5	−321	79
2e	ARMA(1,2), all months	2418	84.8	−323	79
2f	ARMA(2,2), all months	2414	85.5	−321	79

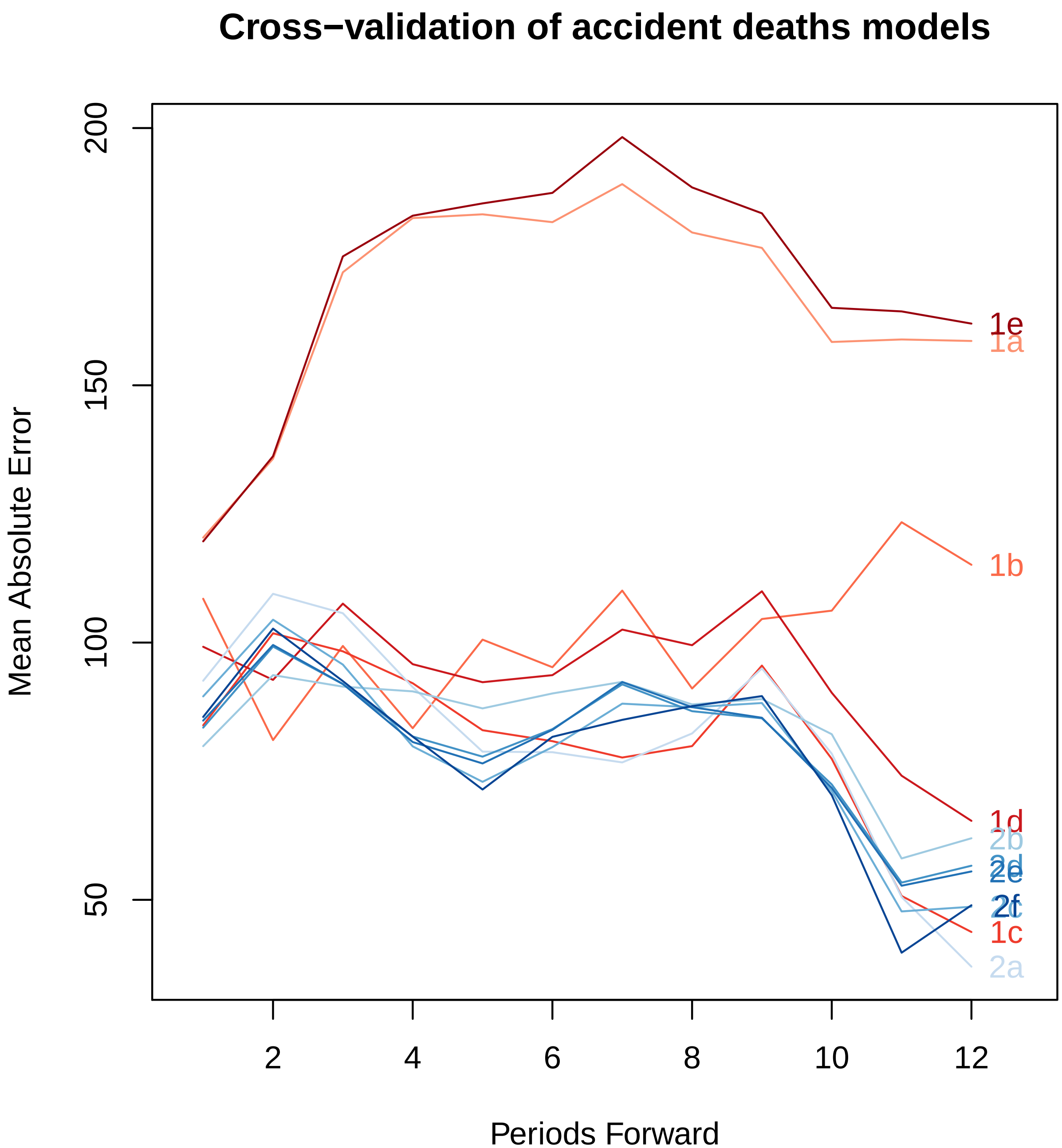
Is MA(1) really the best fitting model, against the in-sample evidence?

Perhaps we should look at forecasts beyond one period ahead?

A graphic helps. . .



Some models seem easy to reject, but which is/are best?



Do we trust the predictions out at 10–12 months? Why or why not?

Summary of fit

Model	Components	AIC	cv12-MAE	β_{Law}	$\mathrm{se}(\beta_{\mathrm{Law}})$
1a	AR(1)	2585	166.4	−377	108
1b	AR(1), q4	2512	101.5	−396	72
1c	AR(1), all months	2438	80.4	−370	70
1d	AR(1), sep to jan	2455	93.6	−378	70
1e	AR(1)AR(1)$_{12}$	2496	170.7	−348	73
2a	AR(2), all months	2425	81.4	−348	81
2b	MA(1), all months	2482	83.7	−392	46
2c	ARMA(1,1), all months	2418	79.5	−323	83
2d,3a	ARMA(2,1), all months	2417	80.3	−321	79
2e	ARMA(1,2), all months	2418	80.1	−323	79
2f	ARMA(2,2), all months	2414	78.1	−321	79

We might summarize the prior figure with the average MAE averaged over the 12 month forecast

This suggests similar performance for most ARMA models,
except MA(1), which is poorer

Summary of fit

Model	Components	AIC	cv8-MAE	β_{Law}	$\mathrm{se}(\beta_{\mathrm{Law}})$
1a	AR(1)	2585	168.0	-377	108
1b	AR(1), q4	2512	96.2	-396	72
1c	AR(1), all months	2438	87.2	-370	70
1d	AR(1), sep to jan	2455	97.9	-378	70
1e	AR(1)AR(1)$_{12}$	2496	171.7	-348	73
2a	AR(2), all months	2425	89.5	-348	81
2b	MA(1), all months	2482	89.1	-392	46
2c	ARMA(1,1), all months	2418	87.2	-323	83
2d,3a	ARMA(2,1), all months	2417	87.0	-321	79
2e	ARMA(1,2), all months	2418	87.0	-323	79
2f	ARMA(2,2), all months	2414	86.0	-321	79

Even discounting the forecasts past 8 months, model 2f comes out slightly ahead. . .

But all of the models with monthly controls and at least one AR term do roughly equally well

Selected model: ARMA(2,2)

```
Coefficients:
         ar1     ar2     ma1      ma2  intercept
      0.0526  0.8449  0.3497  -0.6503  1625.7793
s.e.  0.0538  0.0413  0.1006   0.0998    61.5565
           law      jan       feb       mar        apr
      -312.2308  86.0931  -91.7482  -43.7677  -154.3960
s.e.    81.8335  40.9421   38.1258   40.4084    36.9053
          may       jun      aug      sep      oct
      -19.6984  -72.8430  17.6629  67.3856  209.8757
s.e.   38.9443   34.4385  34.4299  38.9431   36.8765
          nov       dec
      405.8869  526.1152
s.e.   40.3991   38.0647

sigma^2 estimated as 13794:  log likelihood = -1189.2,  aic = 2414.39
```

We have a model – but what does it mean?

Where does this series go over time, with or without the law?

Counterfactual forecasting

We consider two algorithms for forecasting:

Both assume we have point estimates and the variance covariance matrix of the model parameters, $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\phi}}$, $\hat{\boldsymbol{\rho}}$

Both compute forecast over the next K periods given hypothetical values of the covariates, $\mathbf{x}_{c,t+1}, \ldots, \mathbf{x}_{c,t+k}$

Both forecasts are uncertain due to uncertainty in model parameter estimates

Approach 1: **predicted values** $\tilde{y}_{t+k}$, which include the uncertainty due to shocks, $\varepsilon_{t+1}, \ldots, \varepsilon_{t+K}$

For this approach, we also need the estimated variance of these shocks, $\hat{\sigma}^2$

Approach 2: **expected values** $\hat{y}_{t+k}$, which average over the anticipated shocks

Expected values show the expected path of the outcome over the next K periods, given the counterfactual covariates

Counterfactual forecasting: Predicted Values

1. Start in period t with the observed y_t and $\mathbf{x}_t$;
   choose hypothetical $\mathbf{x}_{c,t+k}$'s for each period $t+1, \ldots, t+k, \ldots, t+K$ forecast.

2. Draw a vector of simulated parameters from their asymptotic distribution:
   $\mathrm{vec}\left(\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\rho}}\right) \sim \mathcal{MVN}\left(\mathrm{vec}\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\rho}}\right), \mathrm{Var}\left(\mathrm{vec}\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\rho}}\right)\right)\right)$.

3. Iterate over the following steps for forecast period k in $1, \ldots, K$:

   (a) Draw a new random shock $\tilde{\varepsilon}_{t+1} \sim \mathcal{N}(0, \hat{\sigma}^2)$.

   (b) Calculate one simulated predicted value, $\tilde{y}_{t+k}$ using

   $$\tilde{y}_{t+k} = \sum_{p=1}^{P} y_{t+k-p}\tilde{\phi}_p + \mathbf{x}_{c,t+k}\tilde{\boldsymbol{\beta}} + \sum_{q=1}^{Q} \tilde{\varepsilon}_{t+k-q}\tilde{\rho}_q + \tilde{\varepsilon}_{t+k}.$$

   This formula uses past values of y and ε,
   which may be simulates from prior iterations of the forecast.

4. Repeat steps 2 and 3 `sims` times to construct `sims` simulated forecasts.
   Summarize these predicted values by means and quantiles (predictive intervals).

Counterfactual forecasting: Expected Values

1. Start in period t with the observed y_t and $\mathbf{x}_t$;
   choose hypothetical $\mathbf{x}_{c,t+k}$'s for each period $t+1, \ldots, t+k, \ldots, t+K$ forecast.

2. Draw a vector of simulated parameters from their asymptotic distribution:
   $\mathrm{vec}\left(\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\rho}}\right) \sim \mathcal{MVN}\left(\mathrm{vec}\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\rho}}\right), \mathrm{Var}\left(\mathrm{vec}\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\rho}}\right)\right)\right)$.

3. Iterate over the following step for forecast period k in $1, \ldots, K$:

   (a) Calculate one simulated expected value of y_{t+k} using

   $$\mathbb{E}\left(\tilde{y}_{t+k}|\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\rho}}, \mathrm{x}_{c,t}, \ldots, \mathbf{x}_{c,t+k}, \mathbf{y}_t\right) = \sum_{p=1}^{P} y_{t+k-p}\tilde{\phi}_p + \mathbf{x}_{c,t+k}\tilde{\boldsymbol{\beta}}.$$

   This formula uses past values of y and ε,
   which may be simulates from prior iterations of the forecast.

4. Repeat steps 2 and 3 `sims` times to construct `sims` simulated forecasts.
   Summarize these expected values by means and quantiles (confidence intervals).

Effect of repealing seatbelt law?

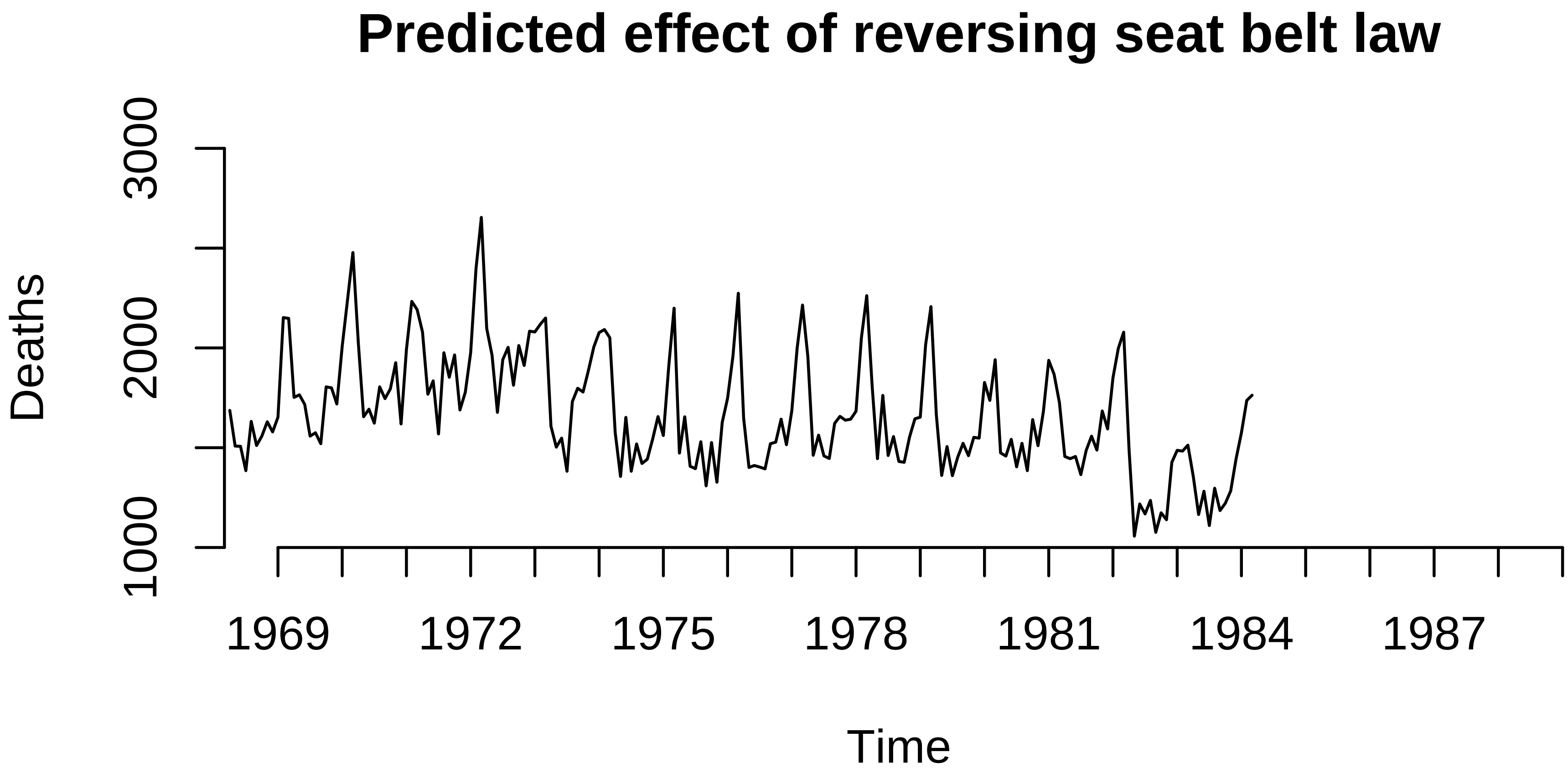
What does the model predict would happen if we repealed the law?

How much would deaths increase after one month? One year? Five years?

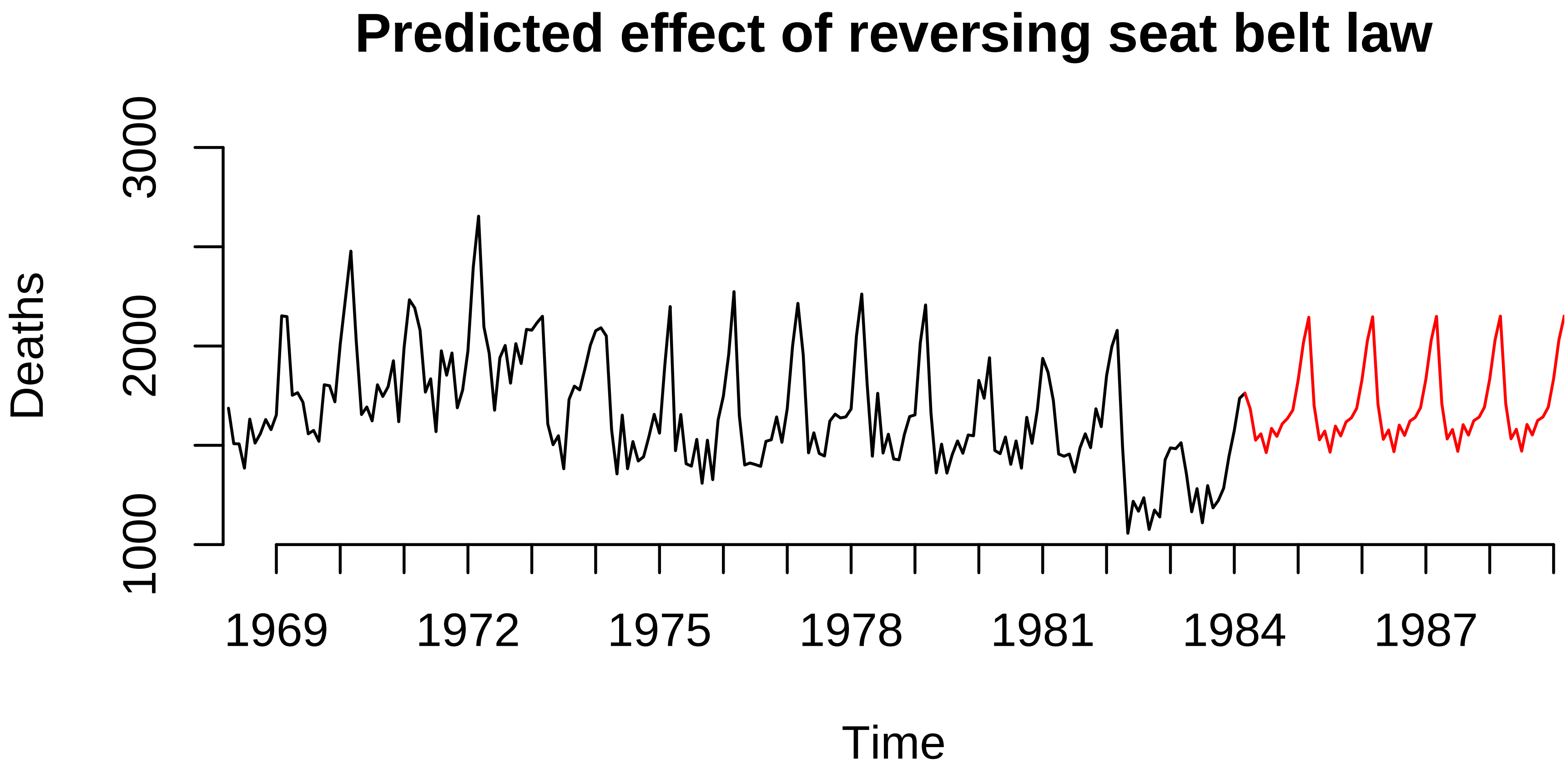
If we run this experiment, how much might the results vary from model expectations?

Need forecast deaths—no law for the next 60 periods, plus predictive intervals

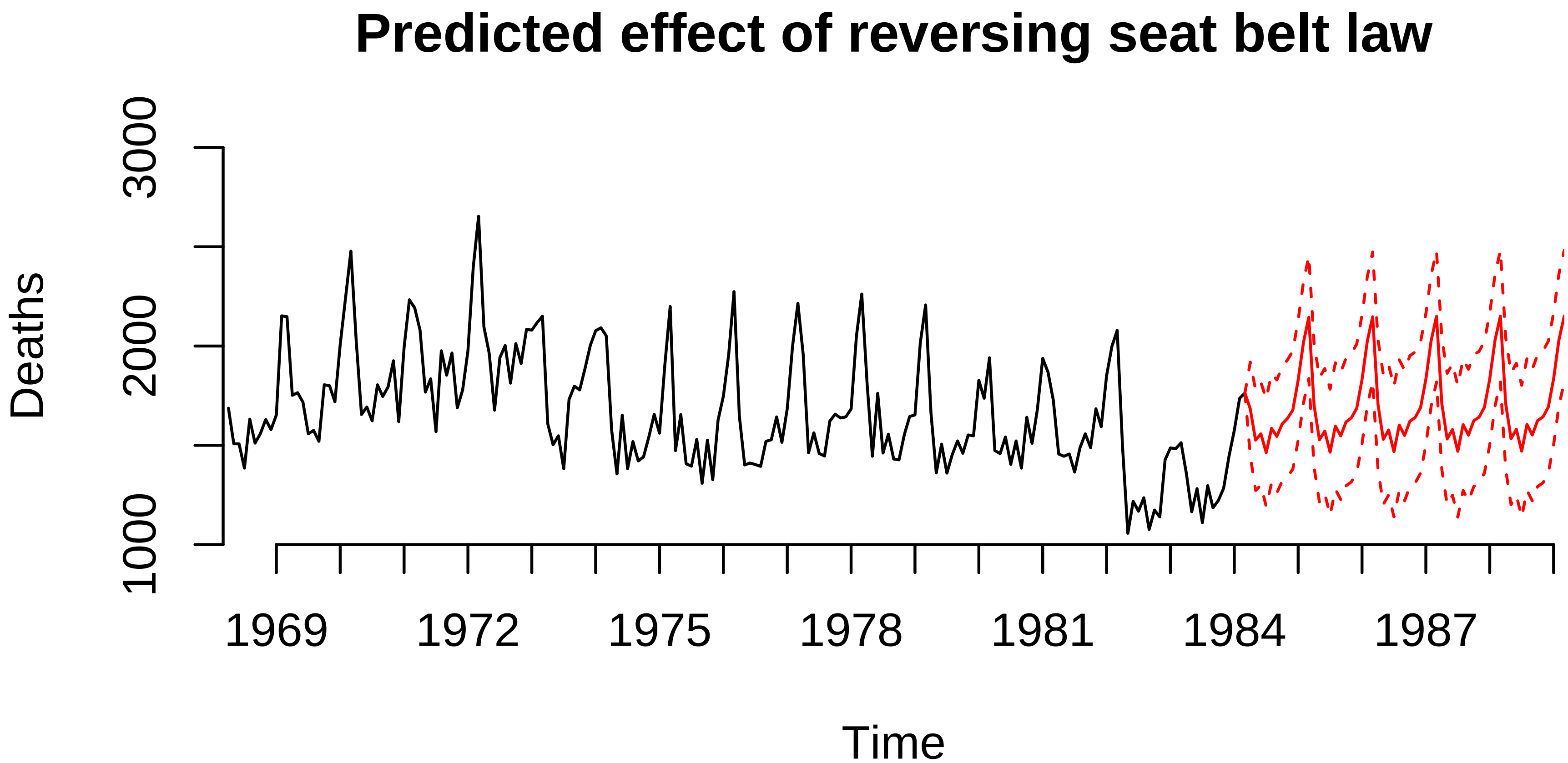
```
predict(arima.res1,                # The model
        n.ahead = 60,              # predict out 60 periods
        newxreg = newdata)         # using these counterfactual x’s
```



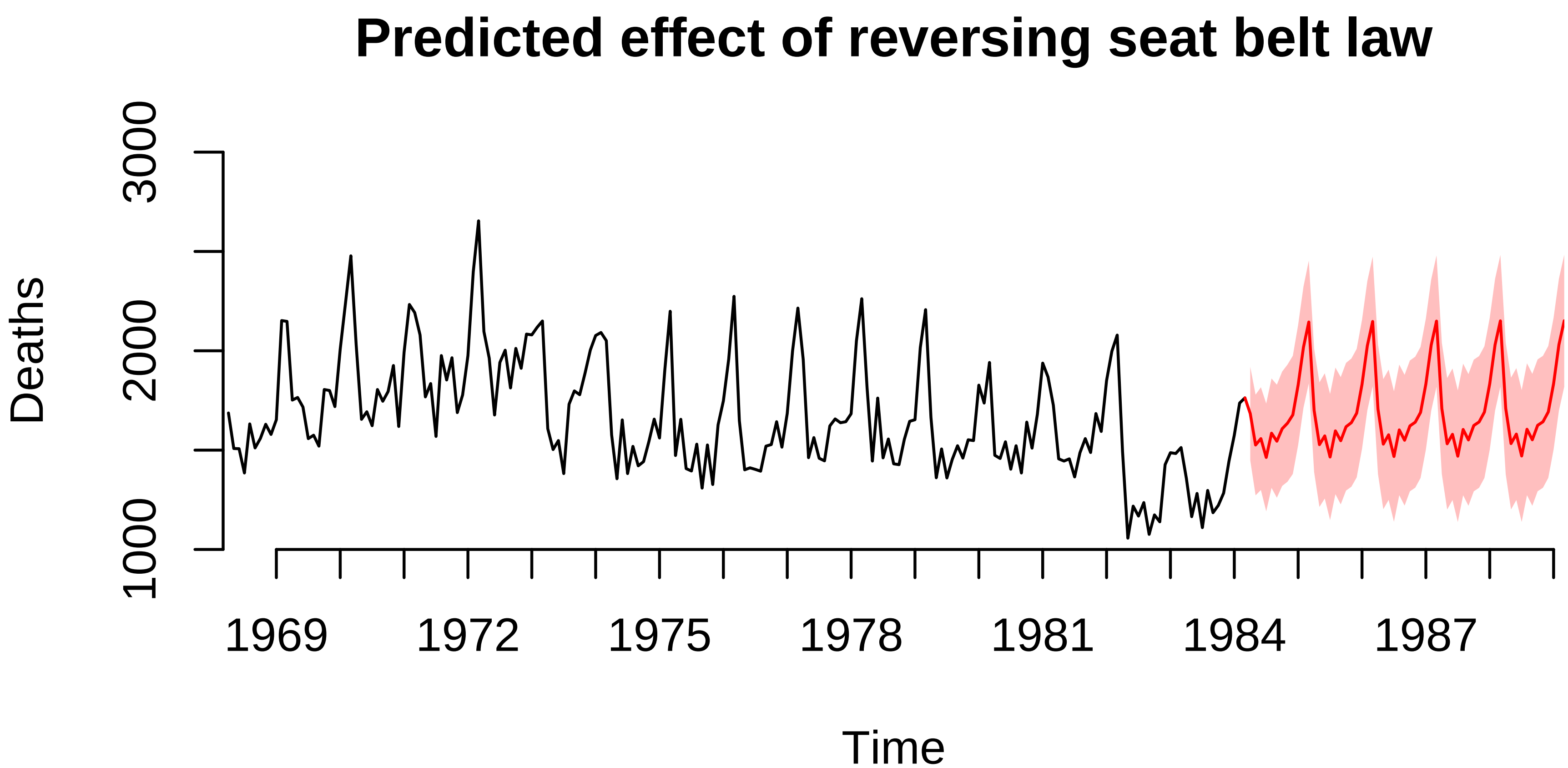
The observed time series



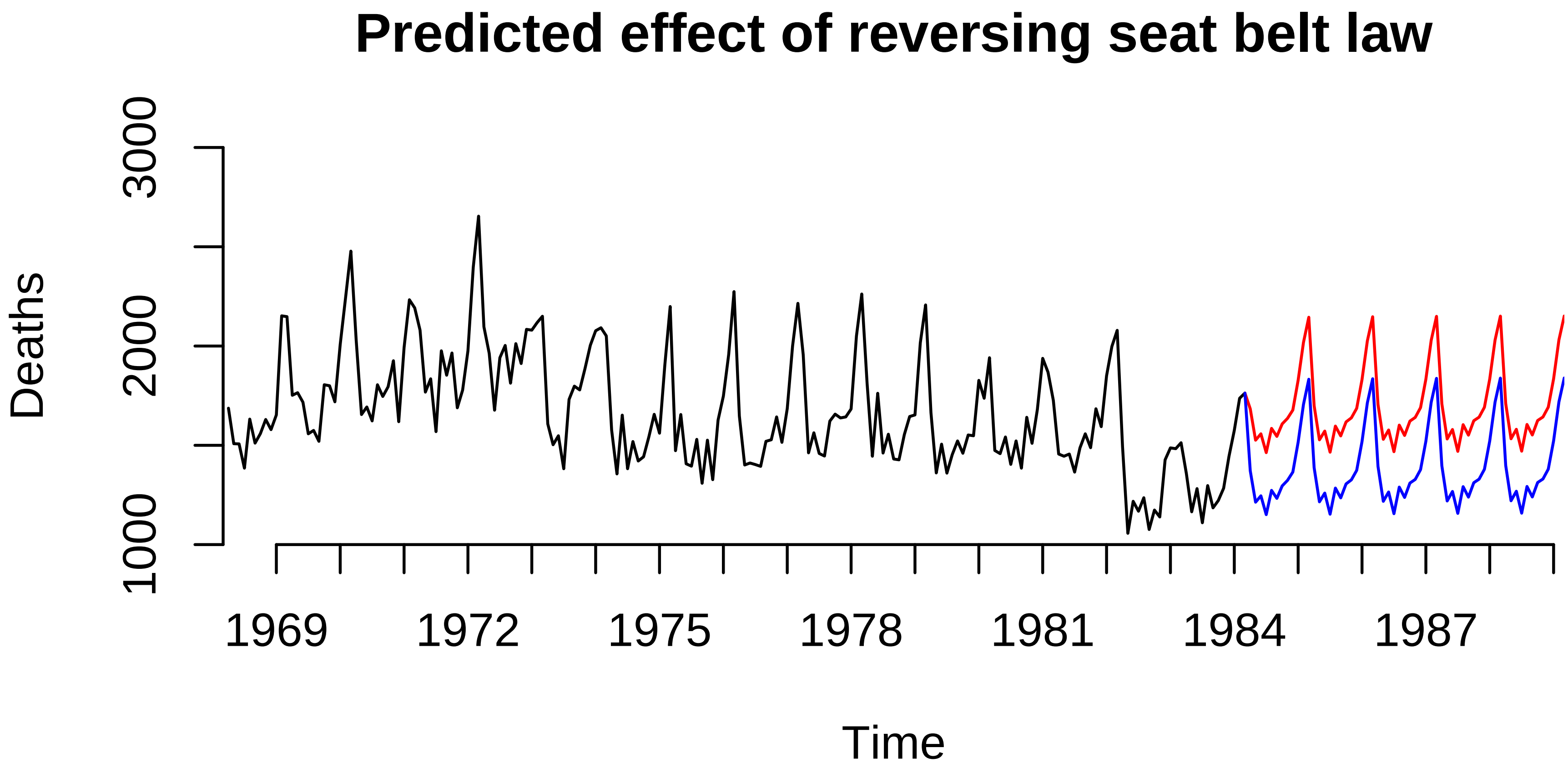
What the model predicts would happen if the seat belt requirement is *repealed*



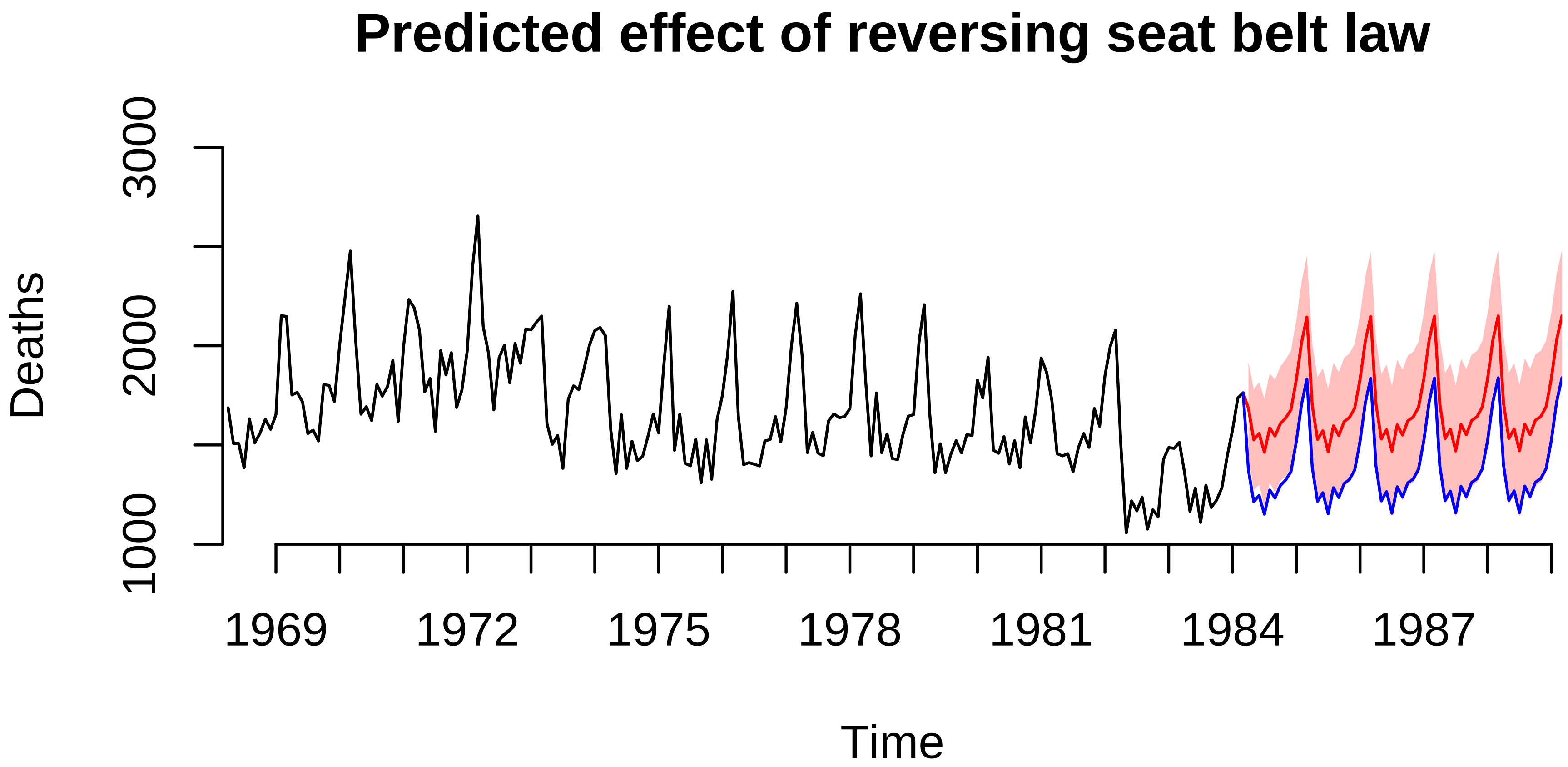
adding the 95 % predictive interval



which is easier to read as a polygon



comparing to what would happen with the law left intact



comparing to what would happen with the law left intact

Confidence intervals vs. Predictive Intervals

Suppose we want *confidence intervals* instead of *predictive intervals*

CIs just show the uncertainty from estimation

Analog to $\mathrm{se}(\boldsymbol{\beta})$ and significance tests

`predict.arima()` *won't* give us CIs

Need to use another package,
or `simcf` (later in the course)

Neat. But is ARMA(p,q) appropriate for our data?

ARMA(p,q) an extremely flexible, broadly applicable model of single time series y_t

But ONLY IF y_t is stationary

If data are non-stationary (have a unit root), then:

- Results may be spurrious
- Long-run predictions impossible

Can assess stationarity through two methods:

1. Examine the data: time series, ACF, and PACF plots
2. Statistical tests for a unit root

Unit root tests: Basic notion

- If y_t is stationary, large negative shifts should be followed by large positive shifts, and vice versa (mean-reversion)

- If y_t is non-stationary (has a unit root), large negative shifts should be uncorrelated with large positive shifts

Thus if we regress $y_t - y_{t-1}$ on y_{t-1}, we should get a negative coefficient if and only if the series is stationary

To do this:

Augmented Dickey-Fuller test `adf.test()` in the `tseries` library

Phillips-Perron test: `PP.test()`

Tests differ in how they model heteroskedasticity, serial correlation,
and the number of lags

Unit root tests: Limitations

Form of unit root test: rejecting the null of a unit root

Will tend to fail to reject for many non-unit roots with high persistence

Very hard to distinguish near-unit roots from unit roots with test statistics

Famously low power tests for single time series

Unit root tests: Limitations

Analogy: Using polling data to predict a very close election

Null Hypothesis: Left Party will get 50.01% of the vote

Alternative Hypothesis: Left will get $< 50\%$ of the vote

We're okay with a 3% CI if we're interested in alternatives like 45% of the vote

But suppose we need to compare the Null to 49.99%

To confidently reject the Null in favor of a very close alternative like this, we'd need a CI of about 0.005% or less

Unit root tests: Limitations

In many political science applications, we ask whether $\phi = 1$ or, say, $\phi = 0.95$

Small numerical difference makes a huge difference for modeling

And single-series unit root tests are weak,
and poorly discriminate across these cases

Simply not much use to us for a single time series,
unless we have panel data

Then we can use panel versions of unit root tests that have somewhat more power

More about panel unit root tests later in the course

Unit root tests: usage

```
> # Check for a unit root
> PP.test(death)

        Phillips-Perron Unit Root Test

data:  death
Dickey-Fuller = -6.435, Truncation lag parameter = 4, p-value = 0.01

> adf.test(death)

        Augmented Dickey-Fuller Test

data:  death
Dickey-Fuller = -6.537, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

Linear regression with y_{t-1} as a control

A popular model in comparative politics & political science is:

$$y_t = y_{t-1}\phi_1 + \mathbf{x}_t\boldsymbol{\beta} + \varepsilon_t$$

estimated by least squares, rather than maximum likelihood

That is, treat y_{t-1} as "just another covariate", rather than a special term

Danger of this approach: y_{t-1} and ε_t are almost certainly correlated (Why?)

Unless we model serial correlation correctly, our errors will be serially correlated, and last period's error is definitely correlated with last period's realization

So if y_{t-1} is treated as a covariate in a linear regression,
this violates G-M condition 2, which requires that $\mathbb{E}(\mathbf{x}_i\varepsilon_i) = 0$

The consequences could be bias in $\hat{\boldsymbol{\beta}}$ and incorrect s.e.'s

When can you use a lag of y as a control in OLS?

My recommendation:

1. Estimate an LS model with the lagged DV
2. Check for remaining serial correlation (Breusch-Godfrey)
3. Compare your results to the corresponding AR(p) estimated by MLE
4. Consider ARMA(p,q) alternatives estimated by MLE
5. Use LS only if it make no statistical or substantive difference

Upshot: You can use LS in cases where it works just as well as MLE

If you model the right number of lags, and need no MA(q) terms,
and have lots of time periods, LS often not far off

Be skeptical of LS standard errors that disgree with AR(p)

Still need to interpret the $\boldsymbol{\beta}$'s and $\boldsymbol{\phi}$'s dynamically

Testing for serial correlation in errors

In LS models, serial correlation makes estimates inefficient (like heteroskedasticity)

If the model includes a lagged dependent variable,
serial correlation $\rightarrow$ inconsistent estimates: $\mathbb{E}(x\varepsilon) \neq 0$

So we need to be able to test for serial correlation.

A general test that will work for single time series or panel data is based on the Lagrange Multiplier

Called Breusch-Godfrey test, or the LM test

Lagrange Multiplier test for serial correlation

1. Run your time series regression by least squares, regressing

$$y_t = \beta_0 + \beta_1 x_{1t} + \ldots + \beta_k x_{kt} + \phi_1 y_{t-1} + \ldots + \phi_k y_{t-p} + u_t$$

2. Regress (using LS) $\hat{u}_t$ on a constant,
   the explanatory variables $x_1, \ldots, x_k, y_{t-1}, \ldots, y_{t-m}$,
   and the lagged residuals, $\hat{u}_{t-1}, \ldots \hat{u}_{t-m}$

   Be sure to chose $m \leq p$. If you choose $m = 1$, you have a test for 1st degree autocorrelation; if you choose $m = 2$, you have a test for 2nd degree autocorrelation, etc.

3. Compute the test-statistic $(T - m)R^2$, where R^2 is the coefficient of determination from the regression in step 2. This test statistic is distributed χ^2 with m degrees of freedom.

4. Rejecting the null for this test statistic is equivalent to rejecting no autocorrelation.

Regression with lagged DV for Accidents

```
Call:
lm(formula = death ~ lagdeath + jan + feb + mar + apr + may +
    jun + aug + sep + oct + nov + dec + law)

Residuals:
    Min      1Q  Median      3Q     Max
-323.58  -84.45   -3.80   80.97  404.88

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  635.11393   96.64706   6.571 5.38e-10 ***
lagdeath       0.64313    0.05787  11.114  < 2e-16 ***
jan         -302.58936   59.33982  -5.099 8.71e-07 ***
feb         -211.00947   48.46926  -4.353 2.26e-05 ***
mar          -31.82070   47.33602  -0.672 0.502314
apr         -177.52653   47.35870  -3.749 0.000241 ***
may           32.58040   47.55810   0.685 0.494199
jun         -111.47957   47.43316  -2.350 0.019863 *
aug          -33.76181   47.52523  -0.710 0.478393
sep            9.48411   47.61220   0.199 0.842339
oct          114.89374   48.04444   2.391 0.017832 *
```

```
nov          224.81981   50.07068   4.490 1.28e-05 ***
dec          213.09991   54.93824   3.879 0.000148 ***
law         -145.31036   37.36477  -3.889 0.000142 ***
---
Signif. codes:
0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 133.9 on 177 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.802,Adjusted R-squared:  0.7875
F-statistic: 55.17 on 13 and 177 DF,  p-value: < 2.2e-16
```

Tests for serial correlation

```
Breusch-Godfrey test for serial correlation of
order up to 1

data:  lm.res1f
LM test = 11.5457, df = 1, p-value = 0.000679


Breusch-Godfrey test for serial correlation of
order up to 2

data:  lm.res1f
LM test = 11.9843, df = 2, p-value = 0.002498
```

Clear evidence of residual serial correlation

Regression with two lags of DV for Accidents

```
Call:
lm(formula = death ~ lagdeath + lag2death + jan + feb + mar +
    apr + may + jun + aug + sep + oct + nov + dec + law)

Residuals:
    Min      1Q  Median      3Q     Max
-378.22  -88.29   -5.04   89.71  308.44

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  475.12645  103.68324   4.582 8.71e-06 ***
lagdeath       0.47250    0.07332   6.445 1.09e-09 ***
lag2death      0.26362    0.07284   3.619 0.000387 ***
jan         -311.45937   57.62112  -5.405 2.09e-07 ***
feb         -329.58156   57.96856  -5.686 5.37e-08 ***
mar          -68.08737   46.99905  -1.449 0.149212
apr         -152.44095   46.46031  -3.281 0.001248 **
may           25.02334   46.18114   0.542 0.588610
jun          -65.76811   47.71466  -1.378 0.169851
aug           -6.16090   46.72852  -0.132 0.895259
sep           19.68658   46.27238   0.425 0.671032
```

```
oct           130.18618   46.79714   2.782 0.005997 **
nov           249.97112   49.06743   5.094 9.00e-07 ***
dec           235.55993   53.65766   4.390 1.96e-05 ***
law          -111.47166   37.45979  -2.976 0.003336 **
---
Signif. codes:
0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 129.8 on 175 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.8155,Adjusted R-squared:  0.8008
F-statistic: 55.26 on 14 and 175 DF,  p-value: < 2.2e-16
```

Tests for serial correlation

```
Breusch-Godfrey test for serial correlation of
order up to 1

data:  lm.res1g
LM test = 0.6961, df = 1, p-value = 0.4041

> bgtest(lm.res1g,2)

Breusch-Godfrey test for serial correlation of
order up to 2

data:  lm.res1g
LM test = 3.2256, df = 2, p-value = 0.1993
```

Perhaps some weak evidence of residual serial correlation,
but as with other tests, hard to be sure if we need to go beyond AR(2)