CSSS 512: Lab 3

Modeling Stationary Time Series

2018-4-20

Agenda

1. Homework 2
2. Box-Jenkins method
3. Estimation and interpretation of ARMA models
4. Cross-validation and model selection
5. Counterfactual forecasting

Homework 2

What to know:

1. ACF, PACF, plotting, demeaning
2. Unit root tests
3. Estimation and interpretation of ARMA models
4. Cross-validation and model selection
5. Counterfactual forecasting

Box-Jenkins Method

Steps:

1. Study generic forms and properties
2. Study these realizations for an indication of which possibly applies to your data
3. Assess your guess–diagnose and iterate
4. Perform a meta-analysis at the end to determine which specification is best

The Box-Jenkins method assumes that time series are composed by multiple temporal processes. It then performs diagnostics to compare the observed series with generic forms to decide what processes occur in the data

Studying the time series

```r
rm(list=ls())

# Load libraries
library(forecast)     # For auto.arima and cross-validation
library(tseries)      # For unit root tests
library(lmtest)       # For Breusch-Godfrey LM test of serial correlation
library(RColorBrewer) # For nice colors
library(MASS)
library(simcf)

# ARIMA Cross-validation by rolling windows
# Adapted from Rob J Hyndman's code:
# http://robjhyndman.com/hyndsight/tscvexample/
#
# Could use further generalization, e.g. to seasonality
# Careful!  This can produce singularities using categorical covariates
arimaCV <- function(x, order, xreg, include.mean, forward=1, minper=50) {
  require(forecast)
  if (!any(class(x)=="ts")) x <- ts(x)
  n <- length(x)
  mae <- matrix(NA, nrow=n-minper, ncol=forward)
  st <- tsp(x)[1]+(minper-2)
  for(i in 1:(n-minper)) {
    xshort <- window(x, start=st+(i-minper+1), end=st+i)
    xnext <- window(x, start=st+(i+1), end=min(n, st+(i+forward)))
    xregshort <- window(xreg, start=st+(i-minper+1), end=st+i)
    xregnext <- window(xreg, start=st+(i+1), end=min(n, st+(i+forward)))
    fit <- Arima(xshort, order=order, xreg=xregshort, include.mean=include.mean)
    fcast <- forecast(fit, h=length(xnext), xreg=xregnext)
    mae[i,1:length(xnext)] <- abs(fcast[['mean']]-xnext)
  }
  colMeans(mae, na.rm=TRUE)
}
```

Studying the time series

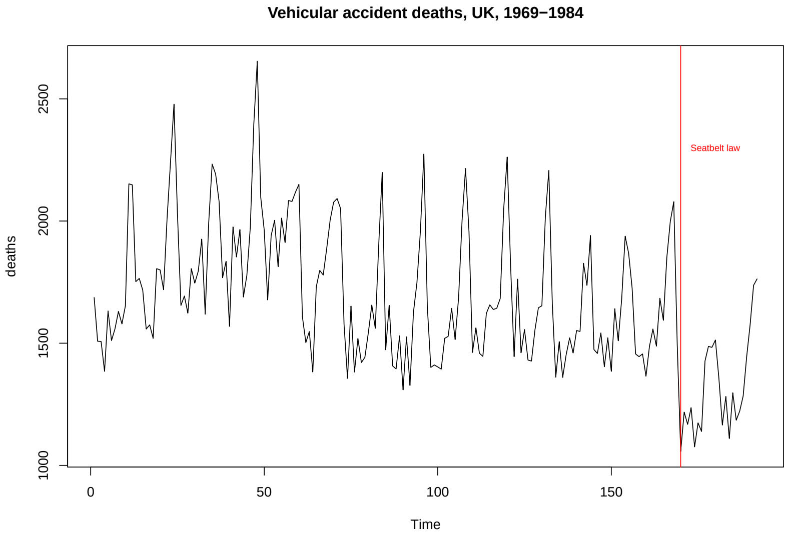
```
# Load data
# Number of deaths and serious injuries in UK road accidents each month.
# Jan 1969 - Dec 1984. Seatbelt law introduced in Feb 1983
# (indicator in second column). Source: Harvey, 1989, p.519ff.
# http://www.staff.city.ac.uk/~sc397/courses/3ts/datasets.html
#
# Variable names:  death law

ukdata <- read.csv("ukdeaths.csv",header=TRUE)
attach(ukdata)

colnames(ukdata)
```

```
## [1] "death" "law"   "year"  "mt"    "month"
```

```
# Look at the time series
```

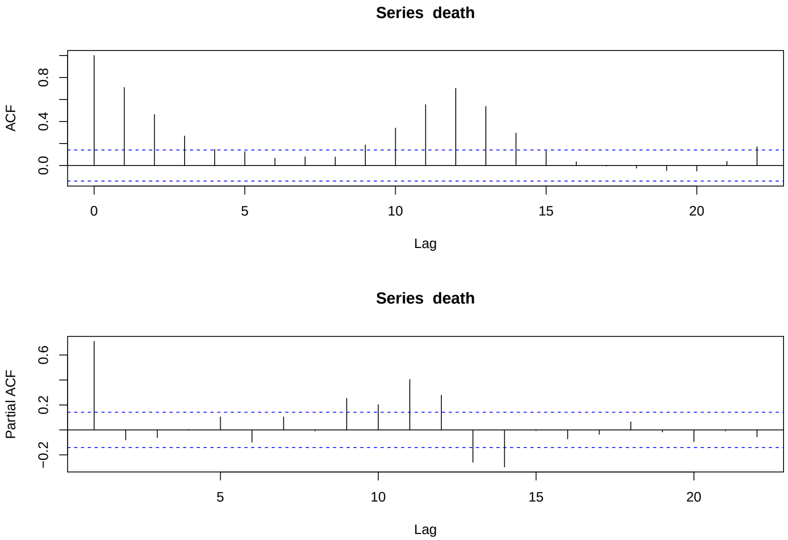
Studying the time series

```
plot(death,type="l",ylab="deaths",xlab="Time", main = "Vehicular accident deaths, UK, 1969-1984")
lines(x=c(170,170),y=c(0,5000),col="red"); text("Seatbelt law",x = 180, y = 2300, col="red",cex=0.7)
```



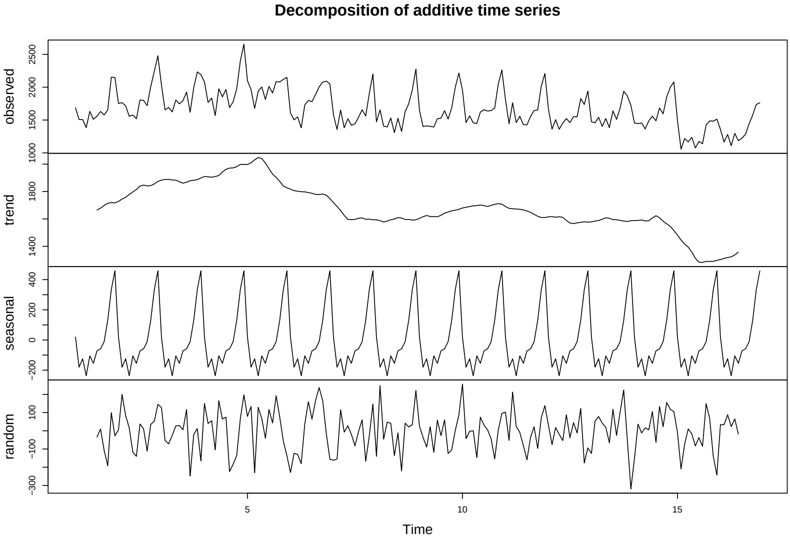
Studying the time series

```r
# Look at the ACF and PACF
par(mfrow=c(2,1))
acf(death); pacf(death)
```



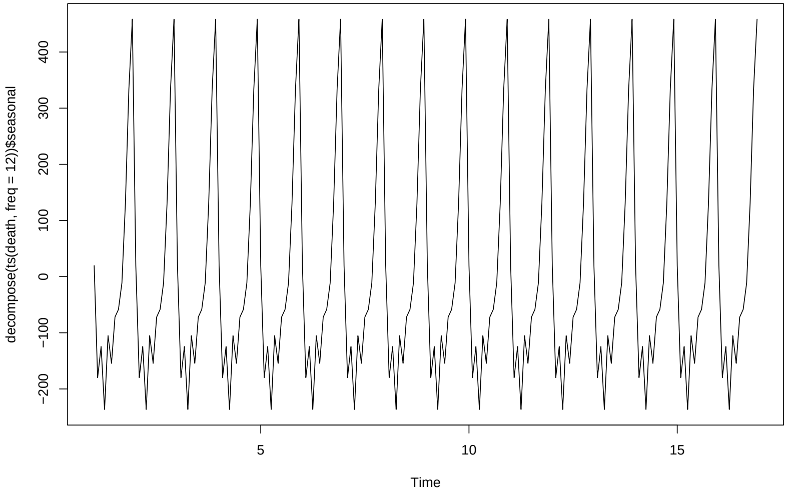
Studying the time series

```
# Look at the decomposed time series
plot(decompose(ts(death,freq=12)))
```



Studying the time series

```
# Look at the monthly cycle
plot(decompose(ts(death,freq=12))$seasonal)
```

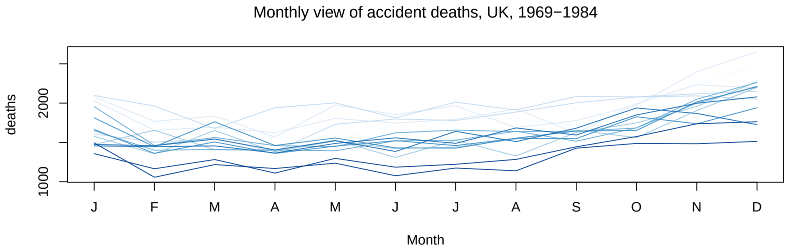


Studying the time series

```r
#Look at the monthly cycle by plotting each year separately

# Make some colors
col <- brewer.pal(8, "Blues")

# Gather the data (sort the number of deaths by month and year in a matrix)
deathmat <- matrix(death,nrow=12,ncol=length(death)/12, byrow=FALSE)

# Repeat them as many times as needed
col <-  as.vector(t(matrix(col, nrow=length(col),
ncol=ceiling(ncol(deathmat)/length(col)))))

# Plot each year over the months
```

Studying the time series

```
par(mfrow=c(2,1))
matplot(deathmat, type="l", col=col, lty=1, xaxt="n", ylab="deaths", xlab="Month",
        main=expression(paste("Monthly view of accident deaths, UK, 1969-1984")))
axis(1, at=1:12, labels=c("J", "F", "M", "A", "M", "J", "J", "A", "S", "O", "N", "D"))
abline(a=0,b=0,lty="dashed")
```



Unit Root Tests

Intuition: if time series is stationary, then regressing $y_t - y_{t-1}$ on y_{t-1} should produce a negative coefficient. Why?

In a stationary series, knowing the past value of the series helps to predict the next period's change. Positive shifts should be followed by negative shifts (mean reversion).

$$y_t = \rho y_{t-1} + \epsilon_t$$

$$y_t - y_{t-1} = \rho y_{t-1} - y_{t-1} + \epsilon_t$$

$$\Delta y_t = \gamma y_{t-1} + \epsilon_t, \text{ where } \gamma = (\rho - 1)$$

Augmented Dickey-Fuller test: null hypothesis of unit root.

Same with Phillips-Perron test, but differs in how the AR(p) time series is modeled: lags, serial correlation, heteroskedasticity.

Unit Root Tests

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

```
##
##  Phillips-Perron Unit Root Test
##
## data:  death
## Dickey-Fuller = -6.4349, Truncation lag parameter = 4, p-value =
## 0.01
```

```r
adf.test(death)
```

```
## Warning in adf.test(death): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  death
## Dickey-Fuller = -6.5366, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
#Do we have evidence that the time series stationary?
```

Studying the time series

```r
#It looks like there is seasonality in the time series, so let's try to control for each month or Q4

# Make some month variables (there are easier ways!)
jan <- as.numeric(month=="January")
feb <- as.numeric(month=="February")
mar <- as.numeric(month=="March")
apr <- as.numeric(month=="April")
may <- as.numeric(month=="May")
jun <- as.numeric(month=="June")
jul <- as.numeric(month=="July")
aug <- as.numeric(month=="August")
sep <- as.numeric(month=="September")
oct <- as.numeric(month=="October")
nov <- as.numeric(month=="November")
dec <- as.numeric(month=="December")

# Make a fourth quarter indicator
q4 <- as.numeric(oct|nov|dec)
```

Studying the time series

```r
# Store all these variables in the dataframe
ukdata$jan <- jan
ukdata$feb <- feb
ukdata$mar <- mar
ukdata$apr <- apr
ukdata$may <- may
ukdata$jun <- jun
ukdata$jul <- jul
ukdata$aug <- aug
ukdata$sep <- sep
ukdata$oct <- oct
ukdata$nov <- nov
ukdata$dec <- dec
ukdata$q4 <- q4

# Set rolling window length and look ahead period for cross-validation
minper <- 170
forward <- 12
```

Estimation and Interpretation

Recall that we use ML to estimate the parameters of an ARMA model (although we also discussed using LS).

This should be familiar:

1. Express the joint probability of the data using the chosen probability distribution
2. Convert the joint probability to the likelihood
3. Simplify the likelihood for easy maximization
4. Substitute the systematic component

Estimation and Interpretation

For an AR(1) process, we are left with the log likelihood function:

$$\mathcal{L}(\boldsymbol{\beta}, \phi_1 | \boldsymbol{y}, \boldsymbol{X}) = -\frac{1}{2}\log\left(\frac{\sigma^2}{1-\phi_1^2}\right) - \frac{\left(y_1 - \frac{\boldsymbol{x_1}\boldsymbol{\beta}}{1-\phi_1}\right)^2}{\frac{2\sigma^2}{1-\phi_1^2}} - \frac{T-1}{2}\log\sigma^2 - \sum_{T=2}^{T}\frac{(y_t - \boldsymbol{x_t}\boldsymbol{\beta} - \phi_1 y_{t-1})^2}{2\sigma^2}$$

We can estimate the parameters, $\boldsymbol{\beta}$ and ϕ_1 that make the data most likely using numerical methods.

The `arima` function in R will compute these for us.

Estimation and Interpretation

```r
## Model 1a:  AR(1) model of death as function of law
##

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

```
##
## Call:
## arima(x = death, order = c(1, 0, 0), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1  intercept  xcovariates
##       0.6439   1719.193    -377.4542
## s.e.  0.0553     42.078     107.6520
##
## sigma^2 estimated as 39289:  log likelihood = -1288.26,  aic = 2584.52
```

```r
#How do we interpret these parameter estimates?

# Extract estimation results from arima.res1a
pe.1a <- arima.res1a$coef                      # parameter estimates (betas)
se.1a <- sqrt(diag(arima.res1a$var.coef))      # standard errors
ll.1a <- arima.res1a$loglik                    # log likelihood at its maximum
sigma2hat.1a <- arima.res1a$sigma2             # standard error of the regression
aic.1a <- arima.res1a$aic                      # Akaike Information Criterion
resid.1a <- arima.res1a$resid                  # residuals
```

Estimation and Interpretation

How do we interpret these parameter estimates?

Introducing the seatbelt law is associated on average with a 378 decrease in the number of road accidents in the next period, all else constant.

Estimation and Interpretation

```r
#Recall that the AIC is equal to the deviance (-2*log likelihood at its maximum) of the model
#plus 2 * the dimension of the model (number of free parameters of the model)
-2*ll.1a + 2*length(pe.1a)
```

```
## [1] 2582.52
```

```r
#And the standard error of the regression is just the expected value of the squared residuals
mean(resid.1a^2)
```

```
## [1] 39289.43
```

```r
#With a fixed mean, where does y_t converge?
```

Estimation and Interpretation

We know in an AR(1) process the effect accumulates over time with a fixed mean.

$$\mathbb{E}(y_t) = \frac{x_t \boldsymbol{\beta}}{1 - \phi_1}$$

We expect the mean level of accidents to converge to

$$\mathbb{E}(y_t) = \frac{x_t \boldsymbol{\beta}}{1 - \phi_1} = \frac{1719.193 - 377.454}{1 - 0.644} = 3768$$

$$\mathbb{E}(y_t) = \frac{x_t \boldsymbol{\beta}}{1 - \phi_1} = \frac{1719.193}{1 - 0.644} = 4828$$

In general, we will forecast these expected values and also predicted values using simulation.

Cross Validation and Model Selection

```
# Attempt at rolling window cross-validation (see caveats)
cv.1a <- arimaCV(death, order=c(1,0,0), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)

cv.1a
```

```
##  [1] 119.6679 136.2173 175.0493 182.9675 185.3571 187.4022 198.2450
##  [8] 188.4625 183.4294 165.0588 164.3636 161.9931
```

Recall that we use a rolling forecast window to perform cross validation.

Describe the steps in your own words.

Estimation and Interpretation

```r
## Model 1b:  AR(1) model of death as function of law & q4

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

```
##
## Call:
## arima(x = 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
```

```r
# Extract estimation results from arima.res1b
pe.1b <- arima.res1b$coef                       # parameter estimates (betas)
se.1b <- sqrt(diag(arima.res1b$var.coef))       # standard errors
ll.1b <- arima.res1b$loglik                     # log likelihood at its maximum
sigma2hat.1b <- arima.res1b$sigma2              # standard error of the regression
aic.1b <- arima.res1b$aic                       # Akaike Information Criterion
resid.1b <- arima.res1b$resid                   # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.1b <- arimaCV(ts(death), order=c(1,0,0), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)

cv.1b
```

```
##  [1] 108.51792  81.04685  99.33850  83.35743 100.56609  95.20655 110.10528
##  [8]  91.06577 104.57805 106.20068 123.39408 115.11284
```

Estimation and Interpretation

```
## Model 1c:  AR(1) model of death as function of law & 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
)
print(arima.res1c)
```

```
##
## Call:
## arima(x = death, order = c(1, 0, 0), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1  intercept        law      jan       feb       mar        apr
##       0.6442  1638.6270  -370.0694  81.3021  -95.1350  -44.3298  -157.3445
## s.e.  0.0550    42.9093    70.2727  54.8127   54.5036   53.0792    50.2149
##           may       jun      aug      sep       oct       nov       dec
##       -19.9428  -75.6674  14.7670  67.4890  206.6686  405.9134  522.0696
## s.e.   45.0247   35.1890  35.1882  45.0184   50.1913   53.0074   54.3054
##
## sigma^2 estimated as 16333:  log likelihood = -1204,  aic = 2437.99
```

```
# Extract estimation results from arima.res1c
pe.1c <- arima.res1c$coef                      # parameter estimates (betas)
se.1c <- sqrt(diag(arima.res1c$var.coef))      # standard errors
ll.1c <- arima.res1c$loglik                    # log likelihood at its maximum
sigma2hat.1c <- arima.res1c$sigma2             # standard error of the regression
aic.1c <- arima.res1c$aic                      # Akaike Information Criterion
resid.1c <- arima.res1c$resid                  # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.1c <- arimaCV(ts(death), order=c(1,0,0), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)
cv.1c
```

```
##  [1]  83.90928 101.80611  98.30016  92.09344  82.96249  80.83424  77.65451
##  [8]  79.87631  95.52618  77.39619  50.74492  43.74215
```

Estimation and Interpretation

```
## Model 1d:  AR(1) model of death as function of law & 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
)
print(arima.res1d)
```

```
##
## Call:
## arima(x = death, order = c(1, 0, 0), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1  intercept        law       jan      sep       oct       nov
##       0.6045  1589.4405  -377.7457  154.7288  80.7422  238.3880  451.3567
## s.e.  0.0575    29.4161    69.7719   35.7336  35.8534   42.6836   44.3474
##           dec
##       579.9770
## s.e.   42.6108
##
## sigma^2 estimated as 18989:  log likelihood = -1218.42,  aic = 2454.83
```

```
# Extract estimation results from arima.res1d
pe.1d <- arima.res1d$coef                          # parameter estimates (betas)
se.1d <- sqrt(diag(arima.res1d$var.coef))          # standard errors
ll.1d <- arima.res1d$loglik                        # log likelihood at its maximum
sigma2hat.1d <- arima.res1d$sigma2                 # standard error of the regression
aic.1d <- arima.res1d$aic                          # Akaike Information Criterion
resid.1d <- arima.res1d$resid                      # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.1d <- arimaCV(ts(death), order=c(1,0,0), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)

cv.1d
```

```
##  [1]  99.18077  92.74232 107.56732  95.79555  92.29416  93.66586 102.52082
##  [8]  99.48355 109.96694  90.24875  74.12182  65.34772
```

Estimation and Interpretation

```r
## Model 1e:  AR(1)AR(1)_12 model of death as function of law
##

## 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
)
print(arima.res1e)
```

```
##
## Call:
## arima(x = 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
```

```r
# Extract estimation results from arima.res1e
pe.1e <- arima.res1e$coef                        # parameter estimates (betas)
se.1e <- sqrt(diag(arima.res1e$var.coef))        # standard errors
ll.1e <- arima.res1e$loglik                      # log likelihood at its maximum
sigma2hat.1e <- arima.res1e$sigma2               # standard error of the regression
aic.1e <- arima.res1e$aic                        # Akaike Information Criterion
resid.1e <- arima.res1e$resid                    # residuals
```

Cross Validation and Model Selection

```
# Attempt at rolling window cross-validation (see caveats)
cv.1e <- arimaCV(ts(death), order=c(1,0,0), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)

cv.1e
```

```
##  [1] 119.6679 136.2173 175.0493 182.9675 185.3571 187.4022 198.2450
##  [8] 188.4625 183.4294 165.0588 164.3636 161.9931
```

```
# So far, an AR(1) with additive seasonality looks best according to AIC
# But maybe a different ARMA(p,q) would fit better?
# Let's keep the additive seasonality and try various ARMA models manually
```

Estimation and Interpretation

```
## Model 2a:  AR(2) model of death as function of law & months

## 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
)
print(arima.res2a)
```

```
##
## Call:
## arima(x = death, order = c(2, 0, 0), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1     ar2  intercept        law      jan       feb       mar
##       0.4696  0.2711  1635.0869  -347.9213  83.7469  -94.9882  -44.0442
## s.e.  0.0692  0.0694    45.6076    80.5683  46.9299   46.5145   45.0452
##             apr       may       jun      aug      sep       oct       nov
##       -157.2316  -19.8376  -75.5957  14.8059  67.5047  206.7362  406.0569
## s.e.    42.8448   37.9719   35.0631  35.0623  37.9640   42.8242   44.9760
##            dec
##       522.4596
## s.e.   46.4368
##
## sigma^2 estimated as 15118:  log likelihood = -1196.65,  aic = 2425.3
```

```
# Extract estimation results from arima.res2a
pe.2a <- arima.res2a$coef                       # parameter estimates (betas)
se.2a <- sqrt(diag(arima.res2a$var.coef))     # standard errors
ll.2a <- arima.res2a$loglik                     # log likelihood at its maximum
sigma2hat.2a <- arima.res2a$sigma2              # standard error of the regression
aic.2a <- arima.res2a$aic                       # Akaike Information Criterion
resid.2a <- arima.res2a$resid                   # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.2a <- arimaCV(ts(death), order=c(2,0,0), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)

cv.2a
```

```
##  [1]  92.57935 109.47089 105.70520  91.33796  78.80066  78.69454  76.70600
##  [8]  82.33373  94.98572  78.40537  50.54681  37.00711
```

Estimation and Interpretation

```r
## Model 2b:  MA(1) model of death as function of law & months

## 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
)
print(arima.res2b)
```

```
##
## Call:
## arima(x = death, order = c(0, 0, 1), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ma1  intercept        law      jan       feb       mar        apr
##       0.4539  1641.4834  -391.7280  79.9732  -94.6320  -44.0097  -157.2155
## s.e.  0.0538    39.7814    45.5288  55.5797   55.6807   55.6807    55.6807
##           may       jun      aug      sep       oct       nov       dec
##       -19.8754  -75.6604  14.8400  67.6897  207.0297  406.5988  522.4457
## s.e.   55.6807   43.9719  43.9719  55.6807   55.6807   55.6807   55.5411
##
## sigma^2 estimated as 20566:  log likelihood = -1225.97,  aic = 2481.93
```

```r
# Extract estimation results from arima.res2b
pe.2b <- arima.res2b$coef                       # parameter estimates (betas)
se.2b <- sqrt(diag(arima.res2b$var.coef))     # standard errors
ll.2b <- arima.res2b$loglik                     # log likelihood at its maximum
sigma2hat.2b <- arima.res2b$sigma2              # standard error of the regression
aic.2b <- arima.res2b$aic                       # Akaike Information Criterion
resid.2b <- arima.res2b$resid                   # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.2b <- arimaCV(ts(death), order=c(0,0,1), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)

cv.2b
```

```
##  [1] 79.86555 93.67862 91.43308 90.51553 87.20853 90.10537 92.36365
##  [8] 87.97704 89.02266 82.18906 58.03778 61.96120
```

Estimation and Interpretation

```
## Model 2c:  ARMA(1,1) model of death as function of law & months

## 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
)
print(arima.res2c)
```

```
##
## Call:
## arima(x = death, order = c(1, 0, 1), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1      ma1  intercept        law      jan       feb       mar
##       0.9349  -0.5994  1629.5549  -323.4929  85.7471  -94.0923  -43.6000
## s.e.  0.0383   0.1076    58.6795    83.2081  40.4544   40.2349   39.7247
##            apr       may       jun      aug      sep       oct       nov
##       -156.8606  -19.6467  -75.5028  14.7339  67.3872  206.5916  405.9572
## s.e.    38.8954   37.7225   36.1673  36.1671  37.7207   38.8896   39.7111
##            dec
##       522.3735
## s.e.   40.2083
##
## sigma^2 estimated as 14568:  log likelihood = -1193.18,  aic = 2418.37
```

```
# Extract estimation results from arima.res2c
pe.2c <- arima.res2c$coef                      # parameter estimates (betas)
se.2c <- sqrt(diag(arima.res2c$var.coef))      # standard errors
ll.2c <- arima.res2c$loglik                    # log likelihood at its maximum
sigma2hat.2c <- arima.res2c$sigma2             # standard error of the regression
aic.2c <- arima.res2c$aic                      # Akaike Information Criterion
resid.2c <- arima.res2c$resid                  # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.2c <- arimaCV(ts(death), order=c(1,0,1), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)
cv.2c
```

```
##  [1]  89.53537 104.43751  95.71992  79.82402  72.93142  79.67158  88.12675
##  [8]  87.42871  88.24699  71.12715  47.74176  48.65514
```

Estimation and Interpretation

```
## Model 2d:  ARMA(2,1) model of death as function of law & months

## 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
)
print(arima.res2d)
```

```
##
## Call:
## arima(x = death, order = c(2, 0, 1), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1      ar2      ma1  intercept        law      jan       feb
##       1.1899  -0.2157  -0.7950  1626.1862  -321.2201  84.8843  -94.5311
## s.e.  0.1071   0.0976   0.0724    68.6982    78.8301  41.3869   41.3010
##           mar        apr       may       jun      aug      sep       oct
##      -43.8782  -157.0544  -19.7871  -75.5646  14.8208  67.5749  206.8634
## s.e.  41.0435    40.5352   39.3222   35.1484  35.1483  39.3216   40.5327
##           nov       dec
##      406.3691  522.9159
## s.e.  41.0341   41.2487
##
## sigma^2 estimated as 14284:  log likelihood = -1191.33,  aic = 2416.66
```

```
# Extract estimation results from arima.res2d
pe.2d <- arima.res2d$coef                     # parameter estimates (betas)
se.2d <- sqrt(diag(arima.res2d$var.coef))     # standard errors
ll.2d <- arima.res2d$loglik                   # log likelihood at its maximum
sigma2hat.2d <- arima.res2d$sigma2            # standard error of the regression
aic.2d <- arima.res2d$aic                     # Akaike Information Criterion
resid.2d <- arima.res2d$resid                 # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.2d <- arimaCV(ts(death), order=c(2,0,1), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)
cv.2d
```

```
##  [1] 83.46358 99.21901 91.91804 81.74041 77.82364 83.17353 91.88016
##  [8] 86.66464 85.28058 72.42659 53.34436 56.62732
```

Estimation and Interpretation

```r
## Model 2e:  ARMA(1,2) model of death as function of law & months

## 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
)
print(arima.res2e)
```

```
##
## Call:
## arima(x = death, order = c(1, 0, 2), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1      ma1      ma2  intercept        law      jan       feb
##       0.9620  -0.5892  -0.1228   1627.146  -322.6854  85.1562  -94.1511
## s.e.  0.0253   0.0752   0.0705     66.814    79.2449  40.7504   40.6400
##           mar        apr       may       jun      aug      sep       oct
##       -43.6591  -156.9126  -19.6915  -75.5237  14.7645  67.4691  206.7084
## s.e.   40.3701    39.9498   39.3736   35.5453  35.5453  39.3730   39.9476
##           nov       dec
##       406.1477  522.6613
## s.e.   40.3650   40.5994
##
## sigma^2 estimated as 14356:  log likelihood = -1191.82,  aic = 2417.63
```

```r
# Extract estimation results from arima.res2e
pe.2e <- arima.res2e$coef                     # parameter estimates (betas)
se.2e <- sqrt(diag(arima.res2e$var.coef))     # standard errors
ll.2e <- arima.res2e$loglik                   # log likelihood at its maximum
sigma2hat.2e <- arima.res2e$sigma2            # standard error of the regression
aic.2e <- arima.res2e$aic                     # Akaike Information Criterion
resid.2e <- arima.res2e$resid                 # residuals
```

Cross Validation and Model Selection

```
# Attempt at rolling window cross-validation (see caveats)
cv.2e <- arimaCV(ts(death), order=c(1,0,2), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)
cv.2e
```

```
##  [1] 84.82861 99.53238 91.96193 80.66522 76.50626 83.03813 92.32656
##  [8] 87.44648 85.38180 71.74271 52.74060 55.51375
```

Estimation and Interpretation

```r
## Model 2f:  ARMA(2,2) model of death as function of law & months

## 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
)
print(arima.res2f)
```

```
##
## Call:
## arima(x = death, order = c(2, 0, 2), xreg = xcovariates, include.mean = TRUE)
##
## Coefficients:
##          ar1     ar2     ma1      ma2  intercept        law      jan
##       0.0526  0.8449  0.3497  -0.6503  1625.7793  -312.2308  86.0931
## s.e.  0.0538  0.0413  0.1006   0.0998    61.5565    81.8335  40.9421
##           feb       mar        apr       may       jun      aug      sep
##       -91.7482  -43.7677  -154.3960  -19.6984  -72.8430  17.6629  67.3856
## s.e.   38.1258   40.4084    36.9053   38.9443   34.4385  34.4299  38.9431
##           oct       nov       dec
##       209.8757  405.8869  526.1152
## s.e.   36.8765   40.3991   38.0647
##
## sigma^2 estimated as 13794:  log likelihood = -1189.2,  aic = 2414.39
```

```r
# Extract estimation results from arima.res2f
pe.2f <- arima.res2f$coef                    # parameter estimates (betas)
se.2f <- sqrt(diag(arima.res2f$var.coef))    # standard errors
ll.2f <- arima.res2f$loglik                  # log likelihood at its maximum
sigma2hat.2f <- arima.res2f$sigma2           # standard error of the regression
aic.2f <- arima.res2f$aic                    # Akaike Information Criterion
resid.2f <- arima.res2f$resid                # residuals
```

Cross Validation and Model Selection

```r
# Attempt at rolling window cross-validation (see caveats)
cv.2f <- arimaCV(ts(death), order=c(2,0,2), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)
cv.2f
```

```
##  [1]  85.54471 102.70679  92.51201  81.72152  71.43657  81.67783  84.97804
##  [8]  87.64424  89.59192  70.33841  39.72134  48.94544
```

Estimation and Interpretation

```
# Tired of manual search?  Auto.arima in the forecast package can automate the search,
# but be careful!
# auto.arima guessed that this time series was non-stationary,
# and without the additive seasonal effects manually added, tried to fit
# complex seasonal ARMA terms. Restricting to a stationary model and
# telling auto.arima to leave seasonality to the month dummies
# produced a simpler, better fitting model
# another warning:  if you seasonal = TRUE, set approximation and stepwise to TRUE
# or prepare to wait
xcovariates <- cbind(law, jan, feb, mar, apr, may, jun, aug, sep, oct, nov, dec)
arima.res3a <- auto.arima(death,
                          stationary=TRUE, seasonal=FALSE,
                          ic="aic", approximation=FALSE, stepwise=FALSE,
                          xreg = xcovariates)
print(arima.res3a)
```

```
## Series: death
## Regression with ARIMA(2,0,1) errors
##
## Coefficients:
##          ar1      ar2      ma1  intercept        law      jan       feb
##       1.1899  -0.2157  -0.7950  1626.1862  -321.2201  84.8843  -94.5311
## s.e.  0.1071   0.0976   0.0724    68.6982    78.8301  41.3869   41.3010
##           mar        apr       may       jun      aug      sep       oct
##      -43.8782  -157.0544  -19.7871  -75.5646  14.8208  67.5749  206.8634
## s.e.  41.0435    40.5352   39.3222   35.1484  35.1483  39.3216   40.5327
##           nov       dec
##      406.3691  522.9159
## s.e.  41.0341   41.2487
##
## sigma^2 estimated as 15582:  log likelihood=-1191.33
## AIC=2416.66   AICc=2420.18   BIC=2472.04
```

Estimation and Interpretation

```r
# Extract estimation results from arima.res3a
pe.3a <- arima.res3a$coef                      # parameter estimates (betas)
se.3a <- sqrt(diag(arima.res3a$var.coef))      # standard errors
ll.3a <- arima.res3a$loglik                    # log likelihood at its maximum
sigma2hat.3a <- arima.res3a$sigma2             # standard error of the regression
aic.3a <- arima.res3a$aic                      # Akaike Information Criterion
resid.3a <- arima.res3a$resid                  # residuals

# Attempt at rolling window cross-validation (see caveats)
cv.3a <- arimaCV(ts(death), order=c(2,0,1), forward=forward,
                 xreg=xcovariates, include.mean=TRUE, minper=minper)
cv.3a
```

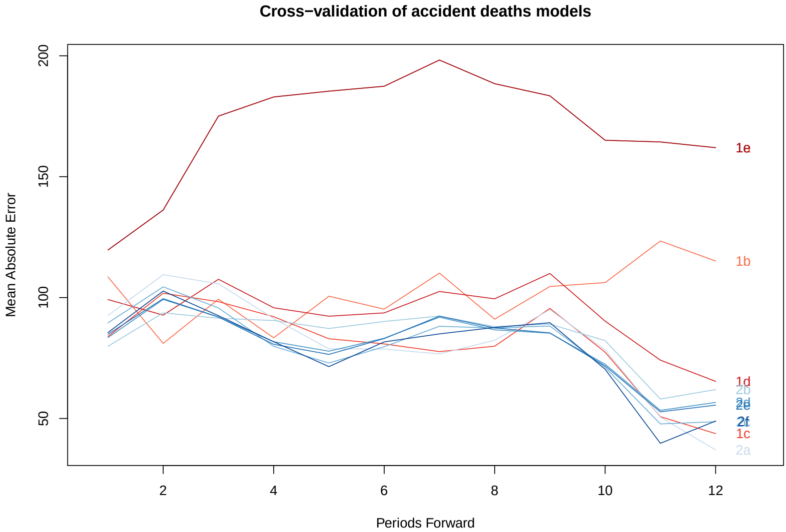
```
##  [1] 83.46358 99.21901 91.91804 81.74041 77.82364 83.17353 91.88016
##  [8] 86.66464 85.28058 72.42659 53.34436 56.62732
```

Cross Validation and Model Selection

```r
# Plot cross-validation results
allCV <- cbind(cv.1a, cv.1b, cv.1c, cv.1d, cv.1e, cv.2a, cv.2b, cv.2c, cv.2d, cv.2e, cv.2f)
labs <- c("1a", "1b", "1c", "1d", "1e", "2a", "2b", "2c", "2d", "2e", "2f")

#pdf("rollingCV.pdf",width=6,height=6.55)

col <- c(brewer.pal(7, "Reds")[3:7],
         brewer.pal(8, "Blues")[3:8])
```

Cross Validation and Model Selection

```r
matplot(allCV, type="l", col=col, lty=1, ylab="Mean Absolute Error", xlab="Periods Forward",
        main="Cross-validation of accident deaths models", xlim=c(0.75,12.75))
text(labs, x=rep(12.5,length(labs)), y=allCV[nrow(allCV),], col=col)
```



Cross Validation and Model Selection

```
# Average cross-validation results
avgCV12 <- sort(apply(allCV, 2, mean))
avgCV8 <- sort(apply(allCV[1:8,], 2, mean))

# Based on AIC & cross-validation,
# let's select Model 2f to be our final model;
# there are other plausible models
arima.resF <- arima.res2f
```

Estimation and Interpretation

```r
## What would happen if we used linear regression on a single lag of death?
lagdeath <- c(NA,death[1:(length(death)-1)])
lm.res1f <- lm(death ~ lagdeath + jan + feb + mar + apr + may + jun +
                 aug + sep + oct + nov + dec + law)
```

Estimation and Interpretation

```
print(summary(lm.res1f))
```

```
##
## 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
```

Estimation and Interpretation

```r
# Check LS result for serial correlation in the first or second order
bgtest(lm.res1f,1)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  lm.res1f
## LM test = 11.546, df = 1, p-value = 0.000679
```

```r
bgtest(lm.res1f,2)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 2
##
## data:  lm.res1f
## LM test = 11.984, df = 2, p-value = 0.002498
```

```r
# Evidence of residual serial correlation is strong
```

Estimation and Interpretation

```r
# Rerun with two lags
lag2death <- c(NA,NA,death[1:(length(death)-2)])
lm.res1g <- lm(death ~ lagdeath + lag2death + jan + feb + mar + apr + may + jun +
                 aug + sep + oct + nov + dec + law)
```

Estimation and Interpretation

```r
print(summary(lm.res1g))
```

```
##
## 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
```

Estimation and Interpretation

Describe in your own words the steps of the Breusch-Godfrey test.

```
# Check LS result for serial correlation in the first or second order
bgtest(lm.res1g,1)
```

```
##
##  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
```

```
# Borderline evidence of serial correlation, but substantively different result.
# (Even small time series assumptions can have big implications for substance.)
# MA terms in ARMA(2,2) seems justified;
# we reject the LS model with lags of the DV
```

Counterfactual Forecasting

Recall that we covered two approaches: forecasting predicted values and forecasting expected values.

How are these different?

Describe the steps in your own words.

Counterfactual Forecasting

```
## Now that we've selected a model, let's interpret it
## using counterfactuals iterated over time
##

## Predict out five years (60 periods) assuming law is kept

# Make newdata dataframe for prediction
xcovariates <- cbind(law, jan, feb, mar, apr, may,
                     jun, aug, sep, oct, nov, dec)
n.ahead <- 60
lawhyp0 <- rep(1,n.ahead)
janhyp0 <- rep( c( 1,0,0, 0,0,0, 0,0,0, 0,0,0 ), 5)
febhyp0 <- rep( c( 0,1,0, 0,0,0, 0,0,0, 0,0,0 ), 5)
marhyp0 <- rep( c( 0,0,1, 0,0,0, 0,0,0, 0,0,0 ), 5)
aprhyp0 <- rep( c( 0,0,0, 1,0,0, 0,0,0, 0,0,0 ), 5)
mayhyp0 <- rep( c( 0,0,0, 0,1,0, 0,0,0, 0,0,0 ), 5)
junhyp0 <- rep( c( 0,0,0, 0,0,1, 0,0,0, 0,0,0 ), 5)
aughyp0 <- rep( c( 0,0,0, 0,0,0, 0,1,0, 0,0,0 ), 5)
sephyp0 <- rep( c( 0,0,0, 0,0,0, 0,0,1, 0,0,0 ), 5)
octhyp0 <- rep( c( 0,0,0, 0,0,0, 0,0,0, 1,0,0 ), 5)
novhyp0 <- rep( c( 0,0,0, 0,0,0, 0,0,0, 0,1,0 ), 5)
dechyp0 <- rep( c( 0,0,0, 0,0,0, 0,0,0, 0,0,1 ), 5)
newdata0 <- cbind(lawhyp0, janhyp0, febhyp0, marhyp0,
                  aprhyp0, mayhyp0, junhyp0, aughyp0,
                  sephyp0, octhyp0, novhyp0, dechyp0)
# Must be in same order as model!
newdata0 <- as.data.frame(newdata0)
names(newdata0) <- c("law", "jan", "feb", "mar", "apr",
                     "may", "jun", "aug", "sep", "oct",
                     "nov", "dec")
```

Counterfactual Forecasting

```
# Run predict
ypred0 <- predict(arima.resF,
                  n.ahead = n.ahead,
                  newxreg = newdata0)

# Simulate predicted values
sims <- 10000
simparam <- mvrnorm(sims, pe.2f, arima.res2f$var.coef)
xhyp <- newdata0
simphi <- simparam[,1:2]
simrho <- simparam[,3:4]
simbetas <- simparam[,5:ncol(simparam)]
lagY <- c(death[length(death)],death[length(death)-1])
lagY <- as.vector(lagY)
lagEps <- c(arima.res2f$resid[length(death)], arima.res2f$resid[length(death)-1])
lagEps <- as.vector(lagEps)
sigma <- sqrt(arima.res2f$sigma)

sim.ev2f <- ldvsimpv(xhyp,
                    simbetas,
                    ci=0.95,
                    constant=1,
                    phi=simphi,
                    lagY=lagY,
                    rho=simrho,
                    lagEps=lagEps,
                    sigma=sigma
)
```

Counterfactual Forecasting

```r
# Predict out five years (60 periods) assuming law is repealed
# Make newdata dataframe for prediction
xcovariates <- cbind(law, jan, feb, mar, apr, may,
                     jun, aug, sep, oct, nov, dec)
n.ahead <- 60
lawhyp <- rep(0,n.ahead)
janhyp <- rep( c( 1,0,0, 0,0,0, 0,0,0, 0,0,0 ), 5)
febhyp <- rep( c( 0,1,0, 0,0,0, 0,0,0, 0,0,0 ), 5)
marhyp <- rep( c( 0,0,1, 0,0,0, 0,0,0, 0,0,0 ), 5)
aprhyp <- rep( c( 0,0,0, 1,0,0, 0,0,0, 0,0,0 ), 5)
mayhyp <- rep( c( 0,0,0, 0,1,0, 0,0,0, 0,0,0 ), 5)
junhyp <- rep( c( 0,0,0, 0,0,1, 0,0,0, 0,0,0 ), 5)
aughyp <- rep( c( 0,0,0, 0,0,0, 0,1,0, 0,0,0 ), 5)
sephyp <- rep( c( 0,0,0, 0,0,0, 0,0,1, 0,0,0 ), 5)
octhyp <- rep( c( 0,0,0, 0,0,0, 0,0,0, 1,0,0 ), 5)
novhyp <- rep( c( 0,0,0, 0,0,0, 0,0,0, 0,1,0 ), 5)
dechyp <- rep( c( 0,0,0, 0,0,0, 0,0,0, 0,0,1 ), 5)
newdata <- cbind(lawhyp, janhyp, febhyp, marhyp,
                 aprhyp, mayhyp, junhyp, aughyp,
                 sephyp, octhyp, novhyp, dechyp)
# Must be in same order as model!
newdata <- as.data.frame(newdata)
names(newdata) <- c("law", "jan", "feb", "mar", "apr",
                    "may", "jun", "aug", "sep", "oct",
                    "nov", "dec")
```

Counterfactual Forecasting

```r
# Run predict
ypred <- predict(arima.resF,
                 n.ahead = n.ahead,
                 newxreg = newdata)

# Simulate predicted values
sims <- 10000
simparam <- mvrnorm(sims, pe.2f, arima.res2f$var.coef)
xhyp <- newdata
simphi <- simparam[,1:2]
simrho <- simparam[,3:4]
simbetas <- simparam[,5:ncol(simparam)]
lagY <- c(death[length(death)],death[length(death)-1])
lagY <- as.vector(lagY)
lagEps <- c(arima.res2f$resid[length(death)], arima.res2f$resid[length(death)-1])
lagEps <- as.vector(lagEps)
sigma <- sqrt(arima.res2f$sigma)

sim.ev2f <- ldvsimpv(xhyp,
                    simbetas,
                    ci=0.95,
                    constant=1,
                    phi=simphi,
                    lagY=lagY,
                    rho=simrho,
                    lagEps=lagEps,
                    sigma=sigma
)
```

Counterfactual Forecasting

```
# Make a plot
pdf("prediction1.pdf",width=6,height=3.25)
plot.new()
par(usr = c(0, length(death) + n.ahead, 1000, 3000) )
# make the x-axis
axis(1,
     at = seq(from = 10, to = 252, by = 12),
     labels = 1969:1989
)
axis(2)

title(xlab = "Time",
      ylab = "Deaths",
      main="Predicted effect of reversing seat belt law")

# Polygon of predictive interval for no law (optional)
x0 <- (length(death)+1):(length(death) + n.ahead)
y0 <- c(ypred$pred - 2*ypred$se, rev(ypred$pred + 2*ypred$se), (ypred$pred - 2*ypred$se)[1] )
polygon(x = c(x0, rev(x0), x0[1]),
        y = y0,
        border=NA,
        col="#FFBFBFFF"
)

# Plot the actual data
lines(x = 1:length(death),
      y = death
)

# Add the predictions for no law
lines(x = length(death):(length(death)+n.ahead),
      y = c(death[length(death)],ypred$pred),  # link up the actual data to the prediction
      col = "red"
)
```