Time Series, Panel Data and Forecasting (Class 9)

Gregory M. Eirich

QMSS

Agenda

1. Lags
   - Distributed lag models
   - Lagged dependent variable
2. “Big T, Small N” panels
3. Co-integration
4. Granger causality

One more point with auto.arima

One can run an *approximate* ARIMAX in auto.arima after all.

It is not exactly the same as an ARIMAX because of the algorithm order it goes through to choose the best ARIMA, but the difference is minimal (and it is probably no worse than trying to just choose the best ARIMA parameters on your own).

Remember this example?

```
> lm.fatal2 <- lm(fatpbvmt ~ unempl + year, data = fatal.unemp)
> summary(lm.fatal2)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 194.800984   5.950313  32.738  < 2e-16 ***
unempl       -0.126555   0.035070  -3.609 0.000616 ***
year         -0.096241   0.003044 -31.618  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4237 on 62 degrees of freedom
Multiple R-squared:  0.9545,   Adjusted R-squared:  0.9531 
F-statistic: 650.8 on 2 and 62 DF,  p-value: < 2.2e-16
```

A negative relationship between unemployment and traffic fatalities, net of trend

ARIMA(0,2,1)

```
> xvars.fat <- fatal.unemp[,c("unempl")]
>
> # ARIMA(1,0,0) = AR(1)
> arima.fat.021 <- arima(fatal.unemp[,"fatpbvmt"], order = c(0,2,1), xreg =
   xvars.fat)
> summary(arima.fat.021)

Call:
arima(x = fatal.unemp[, "fatpbvmt"], order = c(0, 2, 1), xreg = xvars.fat)

Coefficients:
          ma1     xreg
      -0.7994  -0.0866
s.e.   0.1421   0.0145

sigma^2 estimated as 0.01888:  log likelihood = 35.01,  aic = -66.02
```

After some investigation, we settled on ARIMA (0,2,1). Still a negative relationship between unemployment and traffic fatalities, but now in the differenced differences(!)

Simplest auto.arima

Here is just fatalities predicting fatalities. Suggested ARIMA is (1, 2, 1)

```
> m1<-auto.arima(f$fatpbvmt)
> summary(m1)
Series: f$fatpbvmt
ARIMA(1,2,1)

Coefficients:
         ar1      ma1
      0.2719  -0.9405
s.e.  0.1373   0.0527

sigma^2 estimated as 0.02738:  log likelihood=24.52
AIC=-43.03   AICc=-42.63   BIC=-36.56

Training set error measures:
                     ME      RMSE       MAE      MPE     MAPE      MASE
Training set 0.03718118 0.1603807 0.1113402 1.319969 3.258943 0.8395724
                   ACF1
Training set 0.04109592
```

Now, auto.arima-X

Here, we include unemployment. Suggested ARIMA is (2,1,0), with drift. The B is -0.086***, very similar to the ARIMA (0,2,1) from earlier (-0.087***).

```
> m2 = auto.arima(f$fatpbvmt, xreg=f$umempl)
> summary(m2)
Regression with ARIMA(2,1,0) errors

Coefficients:
         ar1     ar2    drift     xreg
      0.2033  0.2189  -0.0991  -0.0864
s.e.  0.1276  0.1277   0.0272   0.0137

sigma^2 estimated as 0.01763:  log likelihood=40.98
AIC=-71.97   AICc=-70.95   BIC=-61.1

Training set error measures:
                     ME      RMSE        MAE       MPE     MAPE      MASE
Training set 0.002050884 0.1276623 0.09547169 0.7126708  3.078241 0.7199142
                    ACF1
Training set 0.001369073
```

auto.arima-X vs. auto.arima on Y alone

It looks like adding unemployment improved prediction, with MAPE (Mean absolute percentage error) dropping to 3.07% vs. 3.26% from the model without that X.

```
> m2 = auto.arima(f$fatpbvmt, xreg=f$umempl)
> summary(m2)
Regression with ARIMA(2,1,0) errors

Coefficients:
         ar1     ar2    drift     xreg
      0.2033  0.2189  -0.0991  -0.0864
s.e.  0.1276  0.1277   0.0272   0.0137


Training set error measures:
                      ME      RMSE        MAE       MPE     MAPE      MASE
Training set 0.002050884 0.1276623 0.09547169 0.7126708  3.078241 0.7199142
                    ACF1
Training set 0.001369073
```

Adding trend too...

Here, we include unemployment + trend. Suggested ARIMA is (0,1,0). The B is -0.10***, quite similar to the Bs of -0.086*** from earlier. (You see that MAPE goes up to 3.83%, so this is a poorer fit.)

```
> xreg <- c(f$year, f$umepl)
> m3 = auto.arima(f$fatpbvmt, xreg=xreg)
> summary(m3)
Regression with ARIMA(0,1,0) errors

Coefficients:
         xreg
      -0.1018
s.e.   0.0207


Training set error measures:
                     ME      RMSE       MAE      MPE     MAPE      MASE
Training set 0.003123124 0.1676516 0.1208714 1.111186  3.831124 0.9114431
                 ACF1
Training set 0.1969959
```

Agenda

1. Lags
   - **Distributed lag models**
   - Lagged dependent variable

The inspiration

A Time Series Analysis of Crime Rates and
Concern for Crime in the United States: 1973-2010

ABSTRACT: Real crime rates may not be the only source of information people use to assess their fear of crime. The present study conducts time series analysis to explore if society does or does not incorporate other information factors into their concern for crime. Using data from the FBI's Uniform Crime Reports and the General Social Survey, I explore the relationship of concern for crime and real crime rates across domains that include covariates of demographic information, national priorities and opinions, and societal values for the years 1973-2010. The study finds support for the argument that people use violent crime rates to logically determine their concern for crime as opposed to using competing sources of information.

Alexandra Vaughn
Columbia University
QMSS 5999 Thesis
Spring 2012

The question

Is there a relationship between published crime rates and the public’s opinions about crime?

A Time Series Problem

- For the United States, from 1973 – 2012
- Try to predict what percent of Americans want to spend more money on crime prevention (from GSS) as a function of:

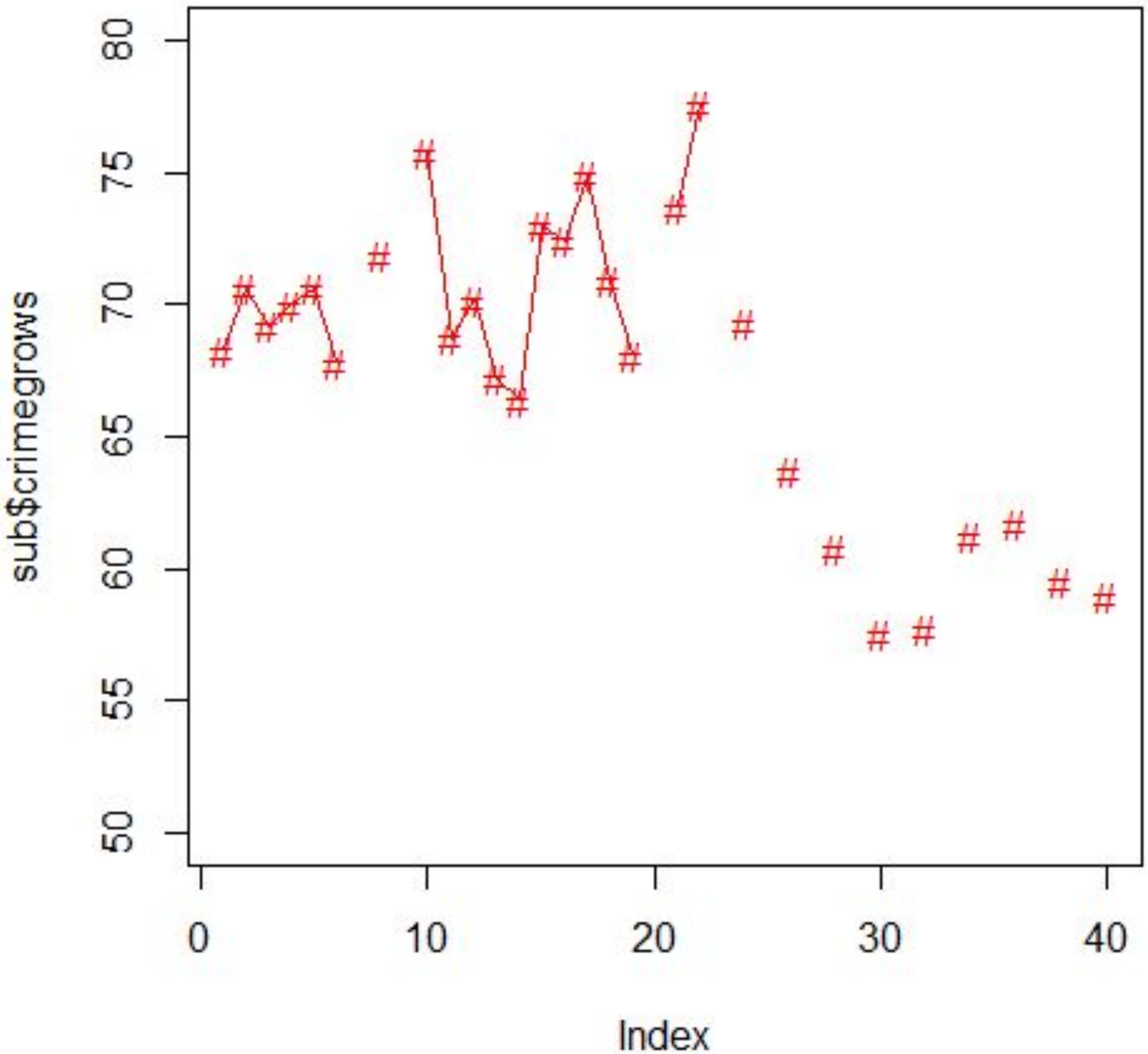
1. The overall violent crime rate for that year
2. And some form of a time trend

** I imputed a lot of the years, especially in the latter half of the series

The missing data

```
   year crimegrows Violent.Crime.rate
1  1973       68.3              417.4
2  1974       70.6              461.1
3  1975       69.2              487.8
4  1976       70.0              467.8
5  1977       70.6              475.9
6  1978       67.8              497.8
7  1979         NA              548.9
8  1980       71.9              596.6
9  1981         NA              593.5
10 1982       75.8              570.8
11 1983       68.7              538.1
12 1984       70.2              539.9
[data omitted]
19 1991       68.1              758.2
20 1992         NA              757.7
21 1993       73.7              747.1
22 1994       77.6              713.6
23 1995         NA              684.5
24 1996       69.3              636.6
25 1997         NA              611.0
26 1998       63.7              567.6
27 1999         NA              523.0
28 2000       60.8              506.5
29 2001         NA              504.5
30 2002       57.5              494.4
31 2003         NA              475.8
32 2004       57.7              463.2
33 2005         NA              469.0
34 2006       61.2              479.3
35 2007         NA              471.8
36 2008       61.7              458.6
37 2009         NA              431.9
```

The original data looks like this



How'd I do that graph?

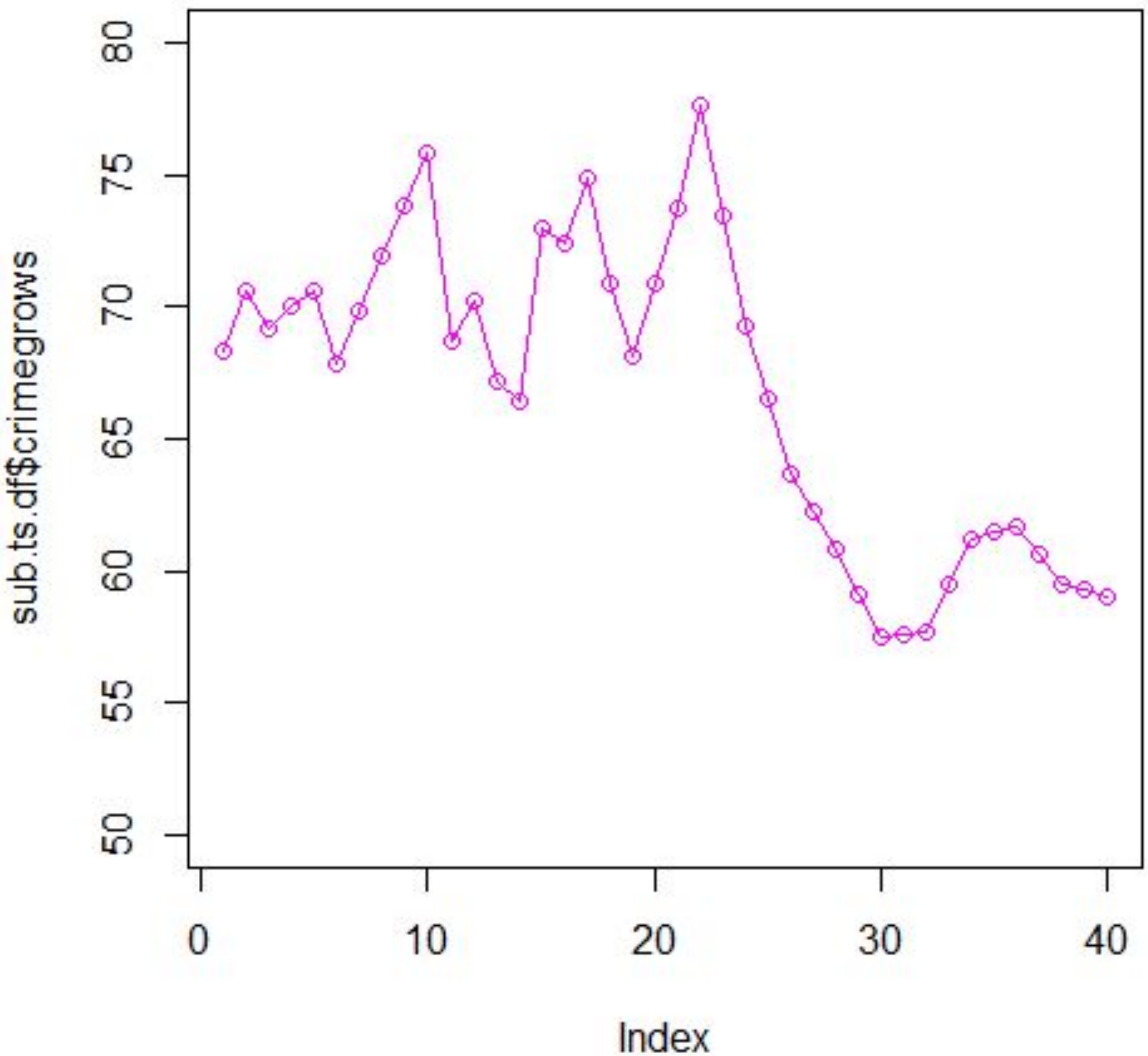
```
total.amelia = read.csv(file.choose())

vars = c("year", "crimegrows", "Violent.Crime.rate")

sub <- total.amelia[, vars]
```

How should I handle missing data?

1. Linearly interpolate: Good at capturing the trended aspect to time series, but might over-determine

2. Impute (singly): Good at adding in natural variance, but can get perhaps unreasonable swings year-to-year

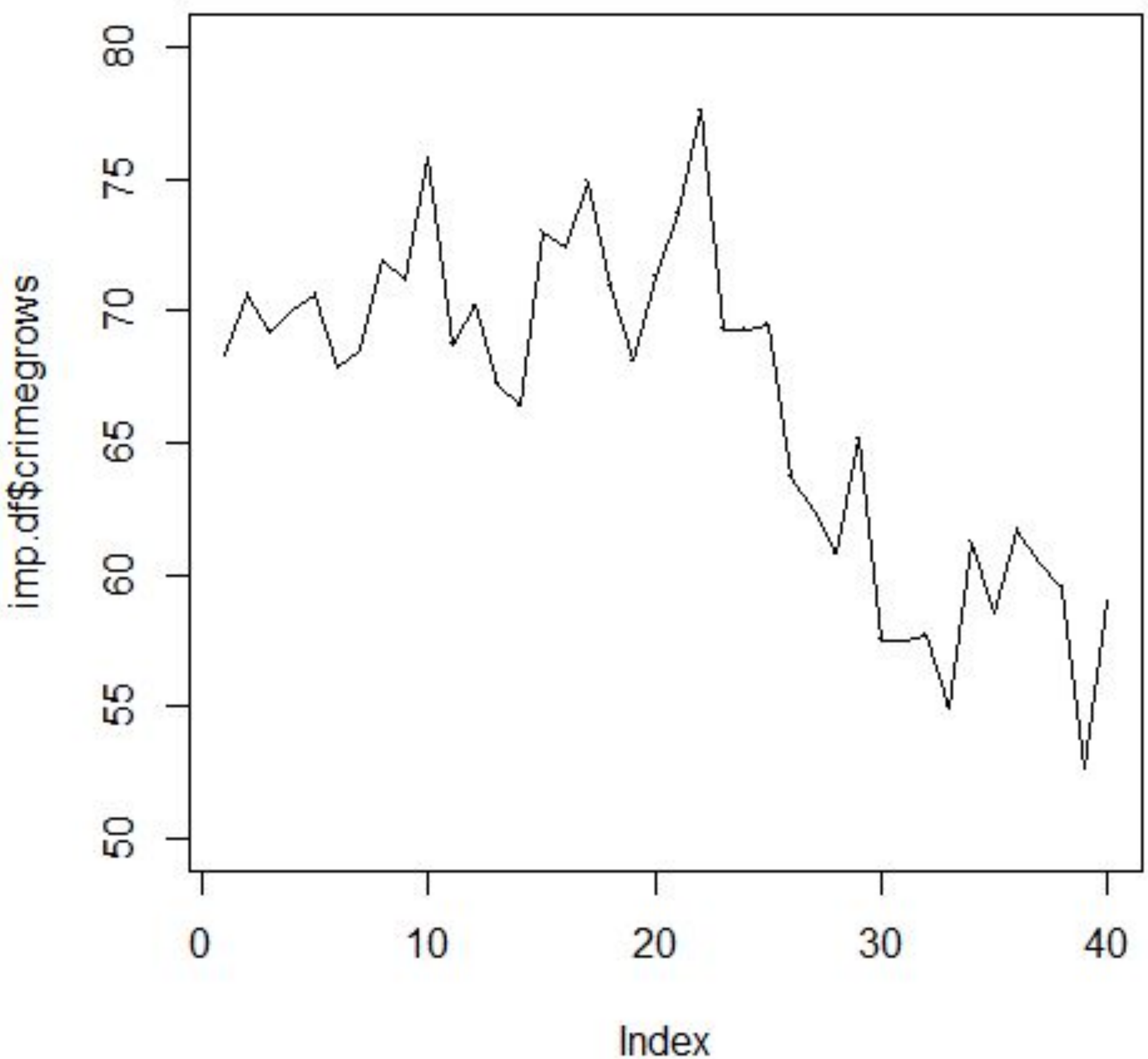
Linear interpolation



How'd I do that graph?

```
sub.ts <- ts(sub)
sub.ts <- na.approx(sub)
sub.ts.df=as.data.frame(sub.ts)
```

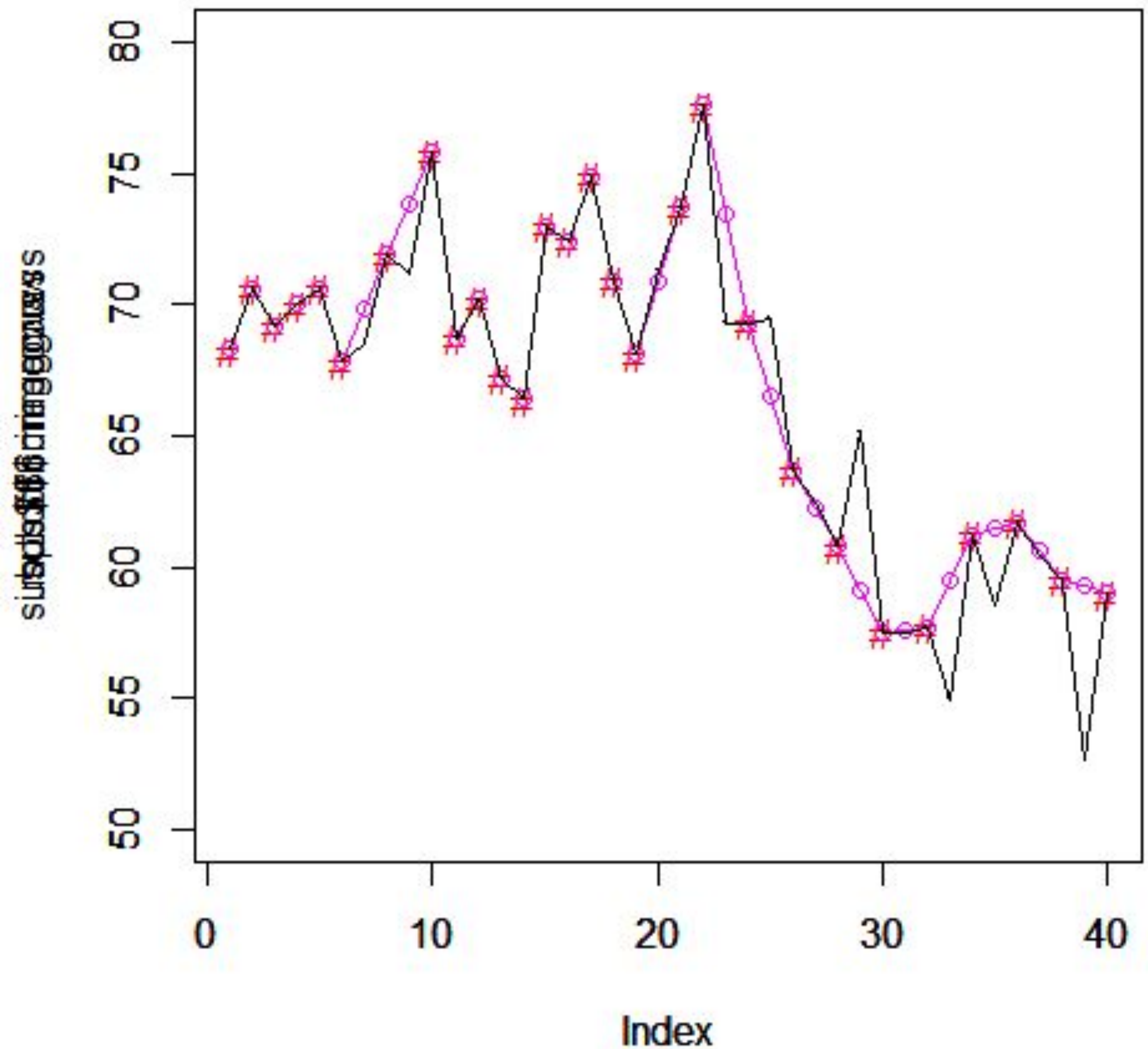
Single imputation



How’d I do that graph?

```
install.packages("Amelia")
library(Amelia)

vars = c("year", "crimegrows", "Violent.Crime.rate")

imp = amelia(total.amelia[,vars], m=1)

imp.df=as.data.frame(imp$imputations)

library(reshape)

imp.df = rename(imp.df, c("imp1.Violent.Crime.rate"="Violent.Crime.rate"))
imp.df = rename(imp.df, c("imp1.crimegrows"="crimegrows"))
imp.df = rename(imp.df, c("imp1.year"="year"))
```

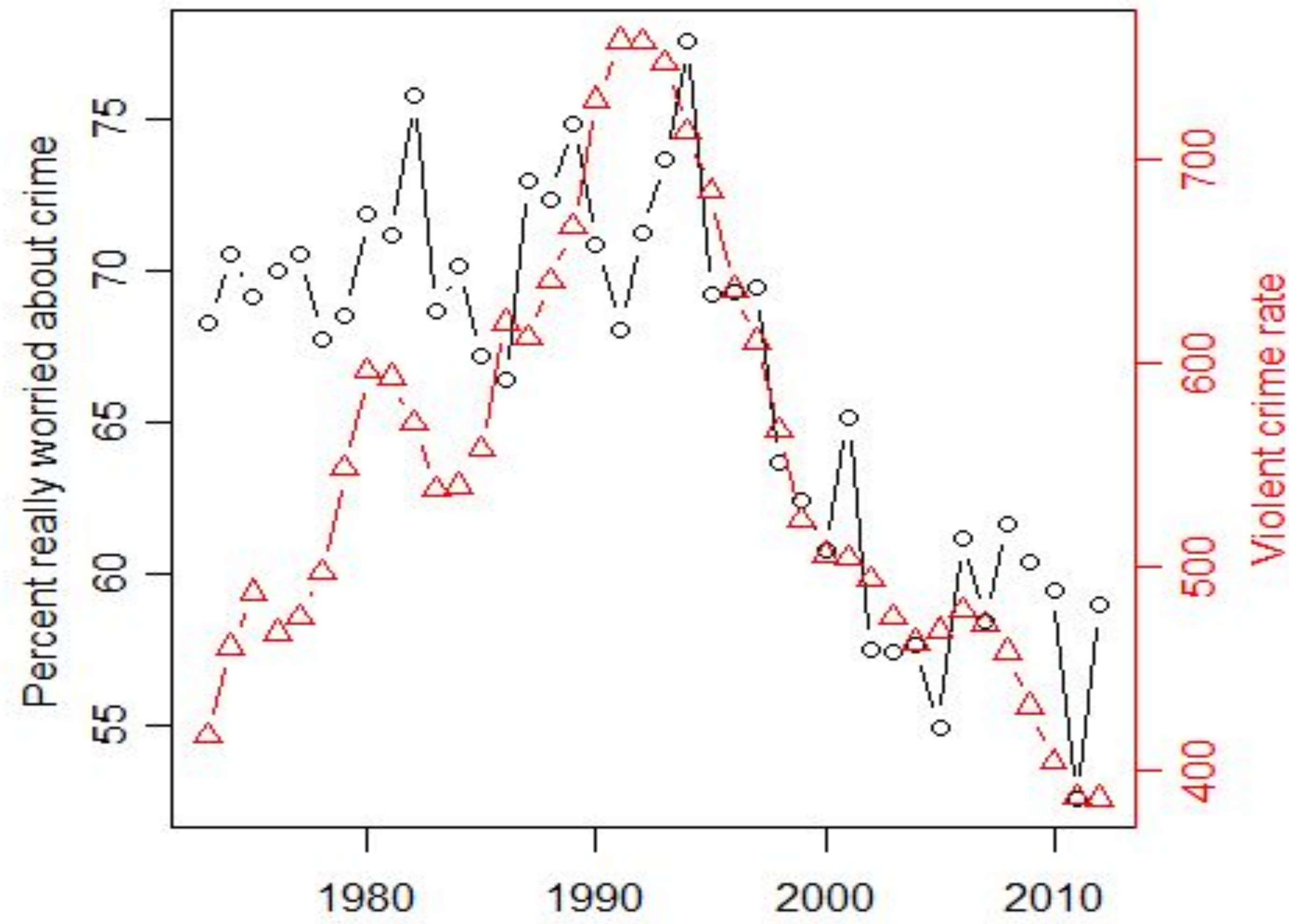
All together



Now, back to our question ...

BTW, we will use the imputed data this time, but feel free to see the difference, if we use another method

Crime perceptions vs. crime rates



How'd I do that graph?

```
install.packages("plotrix")
library(plotrix)

twoord.plot(imp.df$year, imp.df$crimegrows, imp.df$year, imp.df$Violent.Crime.rate,
ylab="Percent really worried about crime", rylab="Violent crime rate")
```

Correlation of variables

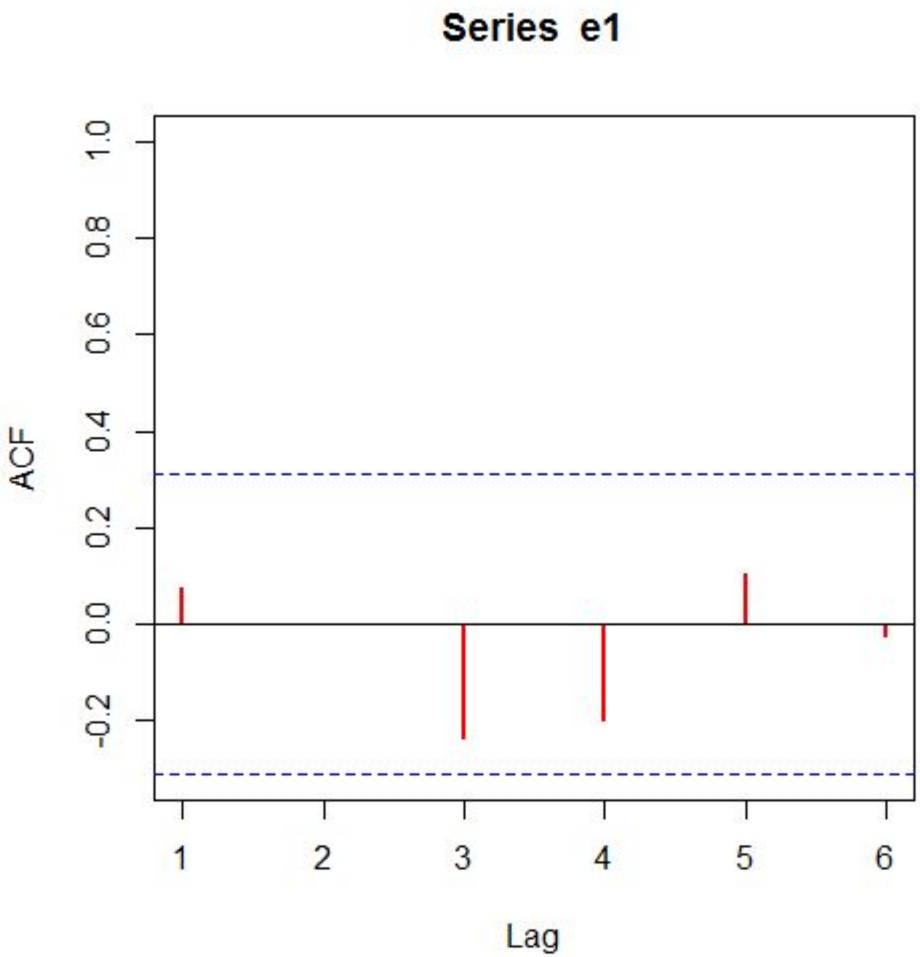
```
> cor.vars <- c("crimegrows", "Violent.Crime.rate", "year")
> cor.dat <- imp.df[, cor.vars]

> cor(cor.dat, use = "complete")
                   crimegrows Violent.Crime.rate       year
crimegrows          1.0000000          0.6766415 -0.7333825
Violent.Crime.rate  0.6766415          1.0000000 -0.2529756
year               -0.7333825         -0.2529756  1.0000000
```

The simplest regression

```
> imp1 = lm(crimegrows ~ Violent.Crime.rate + year, imp.df)
> summary(imp1)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       577.989273  78.111771   7.400 8.38e-09 ***
Violent.Crime.rate   0.026309   0.004335   6.069 5.06e-07 ***
year                -0.263464   0.038883  -6.776 5.65e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.746 on 37 degrees of freedom
Multiple R-squared:  0.7494,  Adjusted R-squared:  0.7358
F-statistic: 55.32 on 2 and 37 DF,  p-value: 7.615e-12
```

For each violent crime/10,000, concern for crime grows by 0.026 percentage points, net of trend

Look for autocorrelation in the errors




```
e1 <- imp1$resid

acf(e1, xlim = c(1,6), col =
"red", lwd = 2)
```

Nothing really going on here …

Any autocorrelation?

```
> Box.test(resid(imp1), lag = 20, type = c("Ljung-Box"), fitdf = 0)

   Box-Ljung test

data:  resid(imp1)
X-squared = 23.8601, df = 20, p-value = 0.2486
```

No sign of AR(1) autocorrelation

The last word on our initial model

```
> e1 <- imp1$resid
> library(forecast)
> auto.arima(e1, trace=TRUE)

 ARIMA(2,0,2) with non-zero mean : Inf
 ARIMA(0,0,0) with non-zero mean : 199.1068
 ARIMA(1,0,0) with non-zero mean : 201.2184
 ARIMA(0,0,1) with non-zero mean : 201.2239
 ARIMA(0,0,0) with zero mean     : 196.8877
 ARIMA(1,0,1) with non-zero mean : 203.6943

 Best model: ARIMA(0,0,0) with zero mean

Series: e1
ARIMA(0,0,0) with zero mean

sigma^2 estimated as 7.627:  log likelihood=-97.39
AIC=196.78   AICc=196.89   BIC=198.47
```

Could not be easier. ARIMA(0,0,0) it is.

Other functional forms for time (or X)?

```
> imp2 = lm(crimegrows ~ Violent.Crime.rate + year + I(year^2), imp.df)
> summary(imp2)
Multiple R-squared:  0.7552,  Adjusted R-squared:  0.7348

> imp.df$late = ifelse(imp.df$year>2001, 1,0)
> imp3 = lm(crimegrows ~ Violent.Crime.rate + late, imp.df)
> summary(imp3)
Multiple R-squared:  0.6543,  Adjusted R-squared:  0.6356

> imp4 = lm(crimegrows ~ Violent.Crime.rate + year + I(year^2) + I(year^3),
   imp.df)
> summary(imp4)
Multiple R-squared:  0.7597,  Adjusted R-squared:  0.7323

> summary(lm(crimegrows ~ Violent.Crime.rate + year + I(Violent.Crime.rate^2),
   imp.df))
Multiple R-squared:  0.7497,  Adjusted R-squared:  0.7288
```

No other specification does better than the linear one for time (= Adj. R-sq = 0.7358)

Question about this model

- Should we think of the relationship between crime rates and crime perception as static and instantaneous? Is that the right way to think about it?

- Is it possible that the relationship is more dynamic, where not just current crime rates can influence perceptions, but past crime rates can also affect perceptions?

Question about this model

- Why would not just current crime rates can influence perceptions, but past crime rates can also affect perceptions:

- Cumulative effects
- Scarring effects

Distributed lag models

- The effects of X on Y are not instantaneous
- Some portion of X's effect is spread over multiple lags of X

Why would lags exist?

- Physical limits
- Psychological limits
- Political limits
- Logistical limits

Lags – while intuitive – can be tricky

- Lose degrees of freedom with each lag

- Heavy multicollinearity among X and lags of X

Getting lags is not so simple

```
vars <- c("year")
date <- imp.df[, vars]
zoo.ts <- zoo(imp.df,date) ## we need to make a zoo time series object ##

by.year.ts.new <- lag(zoo.ts, -4:0, na.pad=T) ## Creates 4 lags of everythings ##

by.year.ts.new.df = as.data.frame(by.year.ts.new)
by.year.ts.new.df

by.year.ts.new.df = rename(by.year.ts.new.df,
   c("Violent.Crime.rate.lag-1"="Violent.Crime.rate.lag1")) ## need to rename these
   because R won't process these names as variables ##
by.year.ts.new.df = rename(by.year.ts.new.df,
   c("Violent.Crime.rate.lag-2"="Violent.Crime.rate.lag2"))
by.year.ts.new.df = rename(by.year.ts.new.df,
   c("Violent.Crime.rate.lag-3"="Violent.Crime.rate.lag3"))
by.year.ts.new.df = rename(by.year.ts.new.df,
   c("Violent.Crime.rate.lag-4"="Violent.Crime.rate.lag4"))
by.year.ts.new.df = rename(by.year.ts.new.df, c("crimegrows.lag-1"="crimegrows.lag1"))


by.year.ts.new.df.ts = ts(by.year.ts.new.df)
```

This generates lags for us

Let me rerun my original model here

```
> lag0 = lm(crimegrows.lag0 ~ Violent.Crime.rate.lag0  + year.lag0  ,
   by.year.ts.new.df.ts)
> summary(lag0)

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             577.989273  78.111771   7.400 8.38e-09 ***
Violent.Crime.rate.lag0   0.026309   0.004335   6.069 5.06e-07 ***
year.lag0                -0.263464   0.038883  -6.776 5.65e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.746 on 37 degrees of freedom
Multiple R-squared:  0.7494,  Adjusted R-squared:  0.7358
F-statistic: 55.32 on 2 and 37 DF,  p-value: 7.615e-12
```

For each violent crime/10,000, concern for crime grows by 0.026 percentage points, net of trend

How about adding 1 lag of X?

```
> lag2 = lm(crimegrows.lag0 ~ Violent.Crime.rate.lag0  +
   Violent.Crime.rate.lag1 + year.lag0  , by.year.ts.new.df.ts)
> summary(lag2)

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)             639.972106  92.489555   6.919 4.86e-08 ***
Violent.Crime.rate.lag0   0.003494   0.017708   0.197    0.845    
Violent.Crime.rate.lag1   0.022564   0.017270   1.307    0.200    
year.lag0                -0.294492   0.046042  -6.396 2.33e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.753 on 35 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.7617,  Adjusted R-squared:  0.7412
F-statistic: 37.28 on 3 and 35 DF,  p-value: 5.356e-11
```

For each violent crime/10,000 from one year ago, concern for crime grows by 0.022 percentage points, net of trend and of this year's crime rate

How about adding 1 lag of X?

```
> vif(lag2)
Violent.Crime.rate.lag0 Violent.Crime.rate.lag1                       year.lag0
              17.009720               15.818402                        1.381801
```

Big multicollinearity ...

Was adding 1 lag of X a good idea?

```
> lag0 = lm(crimegrows.lag0 ~ Violent.Crime.rate.lag0  + year.lag0  ,
   by.year.ts.new.df.ts,   subset= year.lag0>1973)
> summary(lag0)
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             587.867538  84.265970   6.976 3.53e-08 ***
Violent.Crime.rate.lag0   0.025856   0.004587   5.636 2.13e-06 ***
year.lag0                -0.268284   0.041848  -6.411 1.97e-07 ***
---- Multiple R-squared:   0.75,   Adjusted R-squared:  0.7361

> anova(lag0,lag2)

Model 1: crimegrows.lag0 ~ Violent.Crime.rate.lag0 + year.lag0
Model 2: crimegrows.lag0 ~ Violent.Crime.rate.lag0 + Violent.Crime.rate.lag1 +
    year.lag0
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     36 278.19
2     35 265.26  1    12.938 1.7071 0.1999
```

Adding a lag of X does not improve R-sq; though makes maybe some intuitive sense

How about adding another lag of X?

```
> lag3 = lm(crimegrows.lag0 ~ Violent.Crime.rate.lag0  +
   Violent.Crime.rate.lag1 + Violent.Crime.rate.lag2 + year.lag0  ,
   by.year.ts.new.df.ts)
> summary(lag3)

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             619.435342 101.002822   6.133 6.53e-07 ***
Violent.Crime.rate.lag0  -0.002514   0.021562  -0.117    0.908
Violent.Crime.rate.lag1   0.038587   0.036529   1.056    0.298
Violent.Crime.rate.lag2  -0.009999   0.021234  -0.471    0.641
year.lag0                -0.284209   0.050292  -5.651 2.70e-06 ***
---

Residual standard error: 2.82 on 33 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.762,   Adjusted R-squared:  0.7332
F-statistic: 26.42 on 4 and 33 DF,  p-value: 7.014e-10
```

For each violent crime/10,000, concern for crime grows by 0.038 percentage points, net of trend

How about adding another lag of X?

```
> vif(lag3)
Violent.Crime.rate.lag0 Violent.Crime.rate.lag1 Violent.Crime.rate.lag2              year.lag0
              23.536258               64.271098               21.184596               1.453263

> lag0 = lm(crimegrows.lag0 ~ Violent.Crime.rate.lag0  + year.lag0  ,
   by.year.ts.new.df.ts,  subset= year.lag0>1974)

> anova(lag0,lag3)
Analysis of Variance Table

Model 1: crimegrows.lag0 ~ Violent.Crime.rate.lag0 + year.lag0
Model 2: crimegrows.lag0 ~ Violent.Crime.rate.lag0 + Violent.Crime.rate.lag1 +
    Violent.Crime.rate.lag2 + year.lag0
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1     35 278.00
2     33 262.45  2    15.555 0.978 0.3867
```

Not a good idea.

Another example - Wooldridge

In Chapters 10 and 11, we studied various models to estimate the relationship between the general fertility rate (*gfr*) and the real value of the personal tax exemption (*pe*) in the United States. The static regression results in levels and first differences are notably different. The regression in levels, with a time trend included, gives an OLS coefficient on *pe* equal to .187 (se = .035) and R^2 = .500. In first differences (without a trend), the coefficient on Δpe is $-.043$ (se = .028), and R^2 = .032.

Another example, now with lags

In Example 10.4, we explained the general fertility rate, gfr, in terms of the value of the personal exemption, pe. The first order autocorrelations for these series are very large: $\hat{\rho}_1 = .977$ for gfr and $\hat{\rho}_1 = .964$ for pe. These autocorrelations are highly suggestive of unit root behavior, and they raise serious questions about our use of the usual OLS t statistics for this example back in Chapter 10. Remember, the t statistics only have exact t distributions under the full set of classical linear model assumptions. To relax those assumptions in any way and apply asymptotics, we generally need the underlying series to be I(0) processes.

We now estimate the equation using first differences (and drop the dummy variable, for simplicity):

$$\begin{aligned} \widehat{\Delta gfr} &= -.785 - .043\,\Delta pe \\ &\quad\;\; (.502)\;\;(.028) \\ n &= 71,\ R^2 = .032,\ \bar{R}^2 = .018. \end{aligned} \tag{11.26}$$

Now, an increase in pe is estimated to lower gfr contemporaneously, although the estimate is not statistically different from zero at the 5% level. This gives very different results than when we estimated the model in levels, and it casts doubt on our earlier analysis.

If we add two lags of Δpe, things improve:

$$\begin{aligned} \widehat{\Delta gfr} &= -.964 - .036\,\Delta pe - .014\,\Delta pe_{-1} + .110\,\Delta pe_{-2} \\ &\quad\;\; (.468)\;\;(.027) \qquad\;\; (.028) \qquad\quad\;\; (.027) \\ n &= 69,\ R^2 = .233,\ \bar{R}^2 = .197. \end{aligned} \tag{11.27}$$

Even though Δpe and Δpe_{-1} have negative coefficients, their coefficients are small and jointly insignificant (p-value $= .28$). The second lag is very significant and indicates a positive relationship between changes in pe and subsequent changes in gfr two years hence. This makes more sense than having a contemporaneous effect. See Computer Exercise C5 for further analysis of the equation in first differences.

Lots of distributed lag models

Finite distributed lags

- Linear lags
- Polynomial (Almon) lags
- Geometric lags

Infinite Distributed Lags

- Koyck scheme lags
- Rational lags

Lots of distributed lag models

- All of these lags models try to model the lag structure using fewer parameters
- They all involve imposing some sort of structure on the nature of the decay of the lags

Finite distributed lag process w/ diffs

```
> dyn5<-dynlm(d(pray)~d(L(attend,0:3))+d(year), by.year.ts)
> summary(dyn5)

Coefficients: (1 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        0.009169   0.004400   2.084   0.2849
d(L(attend, 0:3))0 0.730460   0.041446  17.624   0.0361 *
d(L(attend, 0:3))1 0.520762   0.040144  12.972   0.0490 *
d(L(attend, 0:3))2 0.441125   0.041862  10.538   0.0602 .
d(L(attend, 0:3))3 0.151735   0.037236   4.075   0.1532
d(year)                  NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.009081 on 1 degrees of freedom
Multiple R-squared:  0.9975,  Adjusted R-squared:  0.9876
F-statistic: 100.3 on 4 and 1 DF,  p-value: 0.07473
```

Very similar story with the difference model. There is a contemporaneous effect and a series of lagged effects.

No unit root, but still AR(1)

```
> auto.arima(resid(dyn3), trace=TRUE)

 ARIMA(2,0,2) with non-zero mean : Inf
 [omitted]
 ARIMA(1,0,1) with non-zero mean : Inf
 ARIMA(2,0,1) with non-zero mean : -9.817379
 ARIMA(1,0,0) with zero mean     : -68.05439
 ARIMA(0,0,0) with zero mean     : -62.98547
 ARIMA(2,0,0) with zero mean     : -62.95428
 ARIMA(1,0,1) with zero mean     : Inf
 ARIMA(2,0,1) with zero mean     : Inf

 Best model: ARIMA(1,0,0) with zero mean

sigma^2 estimated as 1.032e-06:  log likelihood=37.53
AIC=-71.05   AICc=-68.05   BIC=-71.16
```

More to do on this ...

Agenda

1. Lags
   - Distributed lag models
   - **Lagged dependent variable**

Lagged dependent variable

- Usually removes most serial correlation
- Much controversy over this move – can lead to biased and inconsistent estimates
- In time series, the issue is usually whether we think the dynamics are slow- or fast-changing
- In other contexts, this has a nice “causal” interpretation: a change score model (see Wooldridge pp. 310-312)

Lagged dependent variable

```
> arlag1 = lm(crimegrows.lag0 ~ crimegrows.lag1 + Violent.Crime.rate.lag0  +
   year.lag0  , by.year.ts.new.df.ts)
> summary(arlag1)

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)             634.015709 119.968047   5.285 6.77e-06 ***
crimegrows.lag1          -0.087762   0.160810  -0.546    0.589    
Violent.Crime.rate.lag0   0.028131   0.006233   4.513 6.91e-05 ***
year.lag0                -0.289088   0.056915  -5.079 1.26e-05 ***
---
Residual standard error: 2.807 on 35 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.7521,  Adjusted R-squared:  0.7309 
F-statistic:  35.4 on 3 and 35 DF,  p-value: 1.056e-10
```

Net of the public's perception of crime last year and trend, each violent crime/10,000 increases the public's concern over crime by 0.028*** percentage points

What does the ADL model look like?

```
> e2 <- arlag1$resid
> auto.arima(e2, trace=TRUE)

 ARIMA(2,0,2) with non-zero mean : 193.2406
 ARIMA(0,0,0) with non-zero mean : 191.3055
 ARIMA(1,0,0) with non-zero mean : 193.5871
 ARIMA(0,0,1) with non-zero mean : 193.5995
 ARIMA(1,0,1) with non-zero mean : 195.1696
 ARIMA(0,0,0) with zero mean     : 189.0803
 ARIMA(1,0,0) with zero mean     : 191.2348
 ARIMA(0,0,1) with zero mean     : 191.2471
 ARIMA(1,0,1) with zero mean     : 192.6788

 Best model: ARIMA(0,0,0) with zero mean

Series: e2
ARIMA(0,0,0) with zero mean

sigma^2 estimated as 7.073:  log likelihood=-93.49
AIC=188.97   AICc=189.08   BIC=190.64
```

No autocorrelation now.

Agenda

1. Lags
   - Distributed lag models
   - Lagged dependent variable
2. **<u>"Big T, Small N" panels</u>**
3. Co-integration
4. Granger causality

“Big T, Small N” Dataset

- I created a panel dataset of the 9 regions of the United States, from 1975-1992 –
- Try to predict average *imarriedlt50100*, percent of people under 50 who are married:

1. Percent of population under age 50 with at least a BA, *idegree50100*
2. (And time trend, *year*)

I linearly interpolated four of the years (1979, 1981, 1985, 1992), within each region

Some code

```
GSS=read.csv(file.choose())
vars <- c("year", "region", "sex", "age", "marital", "degree")
sub <- GSS[, vars]

# Recodes using mutate from plyr
sub <- mutate(sub,
              married = ifelse(marital == 1, 1, 0),
              baplus = ifelse(degree >= 3, 1, 0),
              marriedlt50 = ifelse(married == 1 & age < 50, 1, 0),
              degreelt50 = ifelse(baplus == 1 & age <50, 1, 0))


# get means by year & region
by.year.region <- aggregate(subset(sub, sel = c(marriedlt50, degreelt50)),
                            by = list(year = sub$year, region = sub$region),
                            FUN = mean, na.rm = T)
```

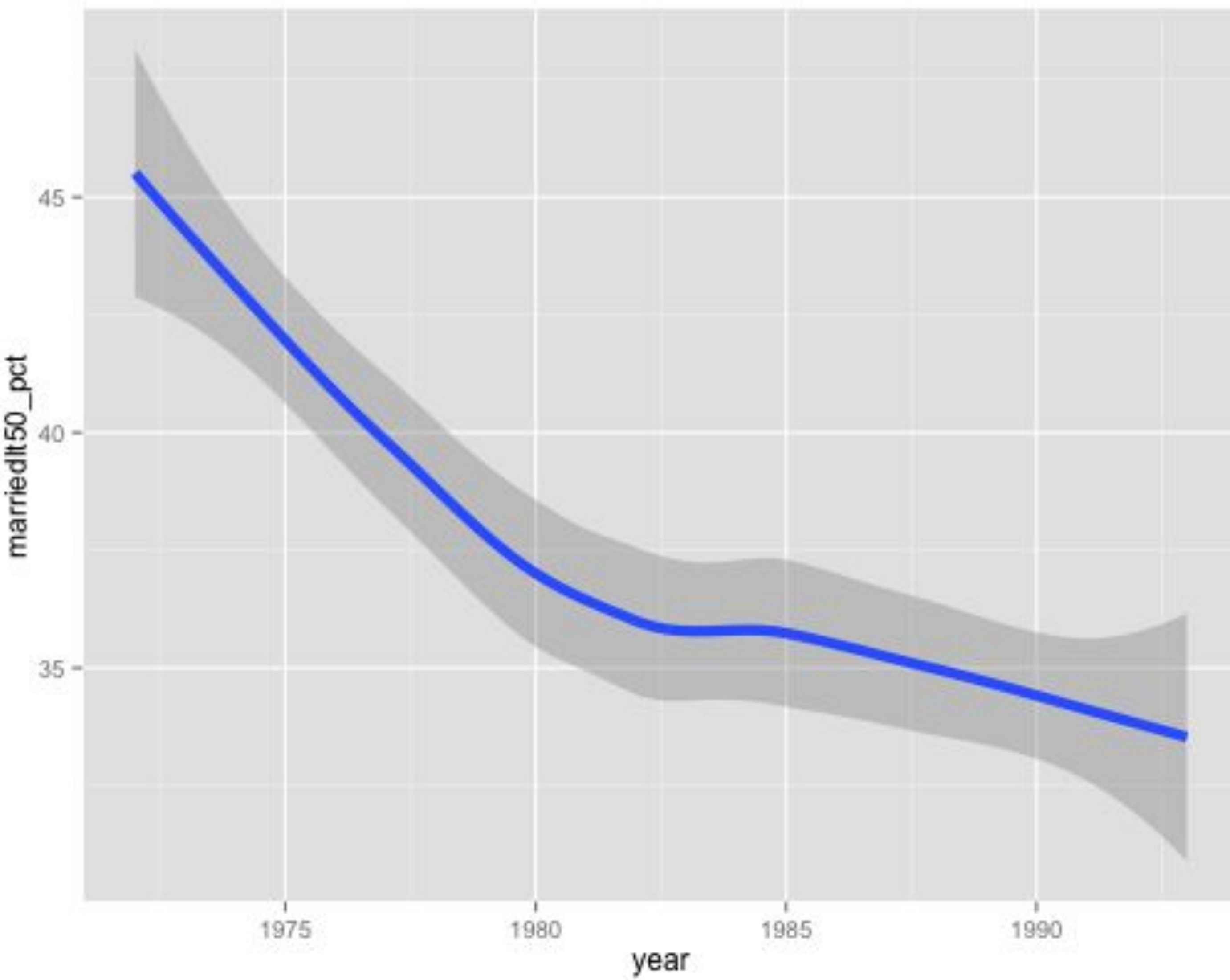
Some code

```
# interpolate for some missing years
interp.dat <- expand.grid(year = c(1979, 1981, 1992), region = 1:9,
                          marriedlt50 = NA, degreelt50 = NA)
by.year.region <- rbind(by.year.region, interp.dat)
by.year.region <- arrange(by.year.region, region, year)
for(i in 1:9){
  sel <- which(by.year.region$region == i)
  temp <- by.year.region[sel,]
  by.year.region[sel, ] <- na.approx(ts(temp))
}


# calculate pct under 50 married, under 50 with BA by year & region
by.year.region <- ddply(by.year.region, c("year", "region"), mutate,
                        marriedlt50_pct = 100*marriedlt50,
                        degreelt50_pct = degreelt50*100)

# only keep up to 1993
by.year.region <- subset(by.year.region, year <= 1993)
```

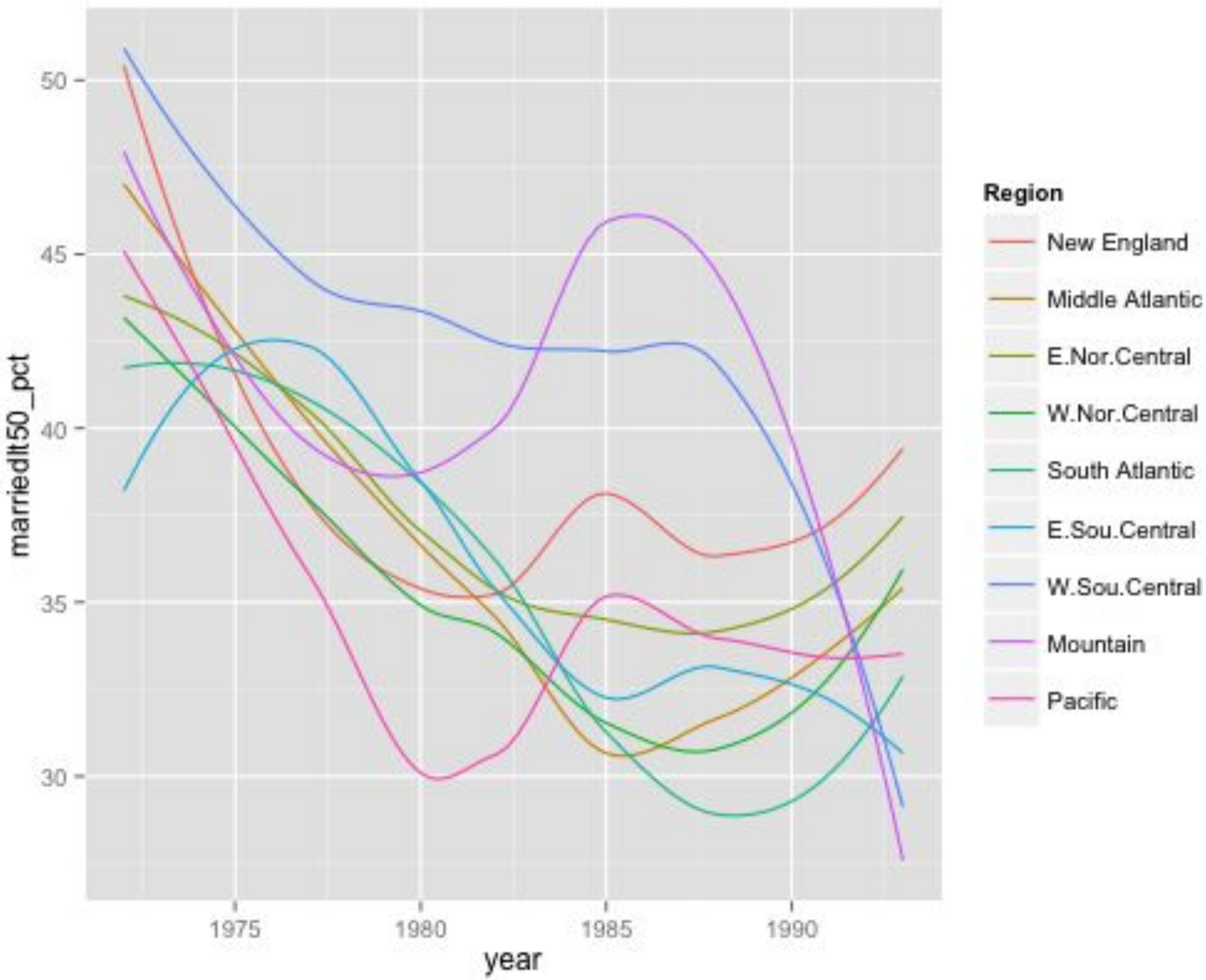
“Big T, Small N” Data

1. Check for serial correlation
2. Check for unit roots
3. Fixed effects? Random effects?
4. AR(1) fixed effects?
5. 1st difference model?

National trend in marriage



Regional trends in marriage



Reminder: The old national regression

```
> lm.married2 <- lm(marriedlt50_pct ~ degreelt50_pct + year, data = by.year.ts)
> summary(lm.married2)

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    2248.7200   289.5368   7.767 3.73e-07 ***
degreelt50_pct    1.6078     0.4198   3.830  0.00123 ** 
year             -1.1253     0.1484  -7.583 5.20e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.524 on 18 degrees of freedom
Multiple R-squared:  0.8808,  Adjusted R-squared:  0.8676 
F-statistic: 66.51 on 2 and 18 DF,  p-value: 4.856e-09
```

Net of the time trend, each percent more of people with BAs *increases* the percent of people married by 1.607 percentage points

Reminder: The old difference model

```
> lm.Dmarried <- lm(firstD(marriedlt50_pct) ~ firstD(degreelt50_pct) + year,
   data = by.year.ts)
> summary(lm.Dmarried)

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -116.10909  127.40211  -0.911  0.37485
firstD(degreelt50_pct)    1.37869    0.39372   3.502  0.00273 **
year                      0.05802    0.06426   0.903  0.37918
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.657 on 17 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.4346,  Adjusted R-squared:  0.3681
F-statistic: 6.535 on 2 and 17 DF,  p-value: 0.007848
```

Net of the time trend, each 1 percentage point difference in the percent of people with BAs increases the percent of people married by 1.378 percentage points

OLS “Big T, Small N” Panel regression

```
> plm.married <- plm(marriedlt50 ~ degreelt50 + as.numeric(year), model =
   "pooling", data = by.year.region)
> clusterSE(plm.married, "region")

t test of coefficients:

                    Estimate   Std. Error t value  Pr(>|t|)
(Intercept)        0.40958312  0.01397723 29.3036 < 2.2e-16 ***
degreelt50         0.29183302  0.13455849  2.1688   0.03131 *
as.numeric(year) -0.00611798  0.00068304 -8.9570 2.674e-16 ***
---
R-Squared      :  0.2835
      Adj. R-Squared :  0.27921
F-statistic: 38.5783 on 2 and 195 DF, p-value: 7.6478e-15
```

Net of the time trend, for every percentage point more of people with BAs there are, the percent of people married increases by .29 percentage points (p<.05)

Serial correlation?

- We cannot look for serial correlation if we have a panel as we have before

OLS “Big T, Small N” Panel regression

```
> plm.married <- plm(marriedlt50 ~ degreelt50 + as.numeric(year) +
   factor(region), model = "pooling", data = by.year.region)
> clusterSE(plm.married, "region")

                     Estimate  Std. Error t value  Pr(>|t|)
(Intercept)        0.38493442  0.02267676 16.9748 < 2.2e-16 ***
degreelt50         0.43462644  0.13810328  3.1471 0.0019192 **
as.numeric(year) -0.00656011  0.00069156 -9.4860 < 2.2e-16 ***
factor(region)2  -0.00232495  0.01489429 -0.1561 0.8761251
factor(region)3   0.01716858  0.01443146  1.1897 0.2356872
factor(region)4  -0.01175962  0.01461530 -0.8046 0.4220664
factor(region)5   0.00039140  0.01764092  0.0222 0.9823222
factor(region)6   0.01400112  0.02131728  0.6568 0.5121190
factor(region)7   0.06850234  0.02013889  3.4015 0.0008195 ***
factor(region)8   0.04278728  0.02056436  2.0807 0.0388284 *
factor(region)9  -0.02079067  0.01336067 -1.5561 0.1213726
---

R-Squared      :  0.43672
      Adj. R-Squared :  0.41246
F-statistic: 14.4986 on 10 and 187 DF, p-value: < 2.22e-16
```

OLS “Big T, Small N” Panel regression

```
> plm.married <- plm(marriedlt50 ~ degreelt50 + as.numeric(year) +
   factor(region), model = "pooling", data = by.year.region)
> clusterSE(plm.married, "region")

                     Estimate  Std. Error t value  Pr(>|t|)
(Intercept)        0.38493442  0.02267676 16.9748 < 2.2e-16 ***
degreelt50         0.43462644  0.13810328  3.1471 0.0019192 **
as.numeric(year) -0.00656011  0.00069156 -9.4860 < 2.2e-16 ***
factor(region)2  -0.00232495  0.01489429 -0.1561 0.8761251
[omitted]
factor(region)7   0.06850234  0.02013889  3.4015 0.0008195 ***
factor(region)8   0.04278728  0.02056436  2.0807 0.0388284 *
factor(region)9  -0.02079067  0.01336067 -1.5561 0.1213726
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Net of the time trend and for each region, every percentage point increase of people with BAs increases the percent of people married by .435 percentage points

F test for the fixed effects model

```
> plm.married <- plm(marriedlt50 ~ degreelt50 + as.numeric(year) +
   factor(region), model = "pooling", data = by.year.region)

> summary(plm.married)$fstatistic

   F test

data:  marriedlt50 ~ degreelt50 + as.numeric(year) + factor(region)
F = 14.4986, df1 = 10, df2 = 187, p-value < 2.2e-16
```

It is highly unlikely that all of these regional dummies are equal to zero (p<.000)

Look for autocorrelation in the errors

```
plm.wood = plm(marriedlt50 ~ degreelt50 + year, index = c("region",
   "year"), model = "within", data = by.year.region)

pbgtest(plm.wood)

   Breusch-Godfrey/Wooldridge test for serial correlation in panel models

data:  marriedlt50 ~ degreelt50 + year
chisq = 69.95, df = 10, p-value = 1.856e-07
alternative hypothesis: serial correlation in idiosyncratic errors
```

Wooldridge (yes, our Wooldridge!) developed a test for serial correlation within panels. We should reject the null of no AR(1) serial correlation. We may have AR(1).

What do we do about this problem?

- You can correct for AR(1) in your panel using the FGLS models, like Cochrane–Orcutt on each panel, but these are not available in R yet, as far as I can tell

- They are in STATA, so let’s see the results ...

An AR(1)-corrected fixed effects model

```
. xi: xtregar  imarriedlt50 idegreelt50, fe

FE (within) regression with AR(1) disturbances  Number of obs      =       189
Group variable (i): region                      Number of groups   =         9
R-sq:  within  = 0.0622                         Obs per group: min =        21
       between = 0.0109                                        avg =      21.0
       overall = 0.0004                                        max =        21

                                                F(1,179)           =     11.88
corr(u_i, Xb)  = -0.2207                        Prob > F           =    0.0007
------------------------------------------------------------------------------
imarriedlt50 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 idegreelt50 |   .3782124   .1097404     3.45   0.001     .1616611    .5947638
       _cons |   .3173206   .0084194    37.69   0.000     .3007064    .3339347
-------------+----------------------------------------------------------------
      rho_ar |  .48452528
     sigma_u |  .02839642
     sigma_e |  .05568742
     rho_fov |  .20636413   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0:     F(8,179) =     1.28               Prob > F = 0.2560
```

An AR(1)-corrected fixed effects model

```
. xi: xtregar  imarriedlt50 idegreelt50, fe

FE (within) regression with AR(1) disturbances  Number of obs      =       189
Group variable (i): region                      Number of groups   =         9
corr(u_i, Xb)  = -0.2207                        Prob > F           =    0.0007
------------------------------------------------------------------------------
imarriedlt50 |      Coef.   Std. Err.       t     P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 idegreelt50 |   .3782124   .1097404     3.45   0.001     .1616611    .5947638
       _cons |   .3173206   .0084194    37.69   0.000     .3007064    .3339347
-------------+----------------------------------------------------------------
      rho_ar |  .48452528
     sigma_u |  .02839642
     sigma_e |  .05568742
     rho_fov |  .20636413   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0:     F(8,179) =     1.28             Prob > F = 0.2560
```

Corrected for AR(1) and for each region, every percentage point increase of people with BAs increases the percent of people married by .378 percentage points

What about random effects?

```
> re.married <- plm(marriedlt50 ~ degreelt50 + as.numeric(year), index =
   c("region", "year"), model = "random", data = by.year.region)
> summary(re.married)
Oneway (individual) effect Random Effect Model
   (Swamy-Arora's transformation)
Effects:
                    var   std.dev share
idiosyncratic 0.0025935 0.0509261   0.8
individual    0.0006487 0.0254705   0.2
theta:  0.6079

Coefficients :
                    Estimate  Std. Error t-value  Pr(>|t|)
(Intercept)       0.39960339  0.01467133 27.2370 < 2.2e-16 ***
degreelt50        0.40447171  0.10524230  3.8432 0.0001642 ***
as.numeric(year) -0.00646674  0.00065714 -9.8408 < 2.2e-16 ***
---
```

Corrected for serial correlation via random effects, every percentage point increase of people with BAs increases the percent of people married by .404 percentage points

Unit roots in panels?

Good resource on unit roots

Testing for Unit Roots: What Should Students Be Taught?

John Elder and Peter E. Kennedy

Abstract: Unit-root testing strategies are unnecessarily complicated because they do not exploit prior knowledge of the growth status of the time series, they worry about unrealistic outcomes, and they double- or triple-test for unit roots. The authors provide a testing strategy that cuts through these complications and so facilitates teaching this dimension of the unit-root phenomenon. *F* tests are used as a vehicle for understanding, but *t* tests are recommended in the end, consistent with common practice.

Key words: teaching econometrics, unit roots

JEL codes: A220, A230, C220

Test that all panels contain unit roots for *imarriedlt50*

```
> summary(purtest(rdat, pmax = 1, test = "levinlin"))
Levin-Lin-Chu Unit-Root Test
Exogenous variables : None
Automatic selection of lags using SIC : 0 - 1 lags (max : 1 )
 Automatic selection of lags using AIC : 0 - 1 lags (max : 1 )
 Automatic selection of lags using Hall : 0 - 1 lags (max : 1 )
statistic : -2.513
p-value : 0.012
   lags obs         rho       trho
X1    0  21 -0.04620787 -1.0637003
X2    1  20 -0.03180087 -1.0922450
X3    1  20 -0.01536668 -0.5232979
X4    0  21 -0.01673521 -0.6272233
X5    1  20 -0.02491039 -1.3312628
X6    1  20 -0.02551853 -0.9315952
X7    0  21 -0.03349647 -0.9109097
X8    1  20 -0.05589454 -1.0360324
X9    0  21 -0.01567761 -0.6895980
```

Based on the Levin-Lin-Chu unit-root test, we can reject the null that **all** panels contain unit roots ($p<.05$)

Test that no panels contain unit roots for *imarriedlt50*

```
> purtest(rdat, pmax = 1, exo = "trend", test = "hadri")

   Hadri Test (ex. var. : Individual Intercepts and Trend )

data:  rdat
z = 5.6857, p-value = 1.303e-08
alternative hypothesis: at least one series has a unit root
```

Based on the Hadri LM test, we can reject the null that all panels are stationary (p<.001)

Individual tests that individual panels are stationary for *imarriedlt50*

```
> urkpssTest(by.year.region[which(by.year.region$region == 1), "marriedlt50"])

Title:
 KPSS Unit Root Test

Test Results:

  Test is of type: mu with 2 lags.

  Value of test-statistic is: 0.2845

  Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
  critical values 0.347 0.463  0.574 0.739
```

Our test-statistic is 0.2845, which is < even 0.347, and so we cannot reject the null of stationarity

… And then do the *KPPS* command for every region …

… And then do the same thing for *idegreelt50* too

Do this test on a loop ...

```
library(fUnitRoots)
urkppsTest(by.year.region[which(by.year.region$region == 1), "marriedlt50"])

# can do it for each region using a loop
for(i in 1:9){
  test <- urkppsTest(by.year.region[which(by.year.region$region == i),
   "marriedlt50"])
  print(paste("Test for Region = ", i))
  print(test)
}
```

What do we conclude?

- We can conclude that we don’t have all unit roots for all individuals, nor do we have no unit roots for every individual …
- So we are somewhat in-between
- We may just want to take the differences to be safe too …
- (But that differencing might induce some serial correlation …)

1st differences "Big T, Small N" regression

```
> plm.fd = plm(marriedlt50 ~ degreelt50 , index = c("region", "year"), model =
   "fd", data = by.year.region)
> summary(plm.fd)
Oneway (individual) effect First-Difference Model

Balanced Panel: n=9, T=22, N=198

Coefficients :
              Estimate Std. Error t-value  Pr(>|t|)
(intercept) -0.0074364  0.0046891 -1.5859    0.1144
degreelt50   0.4885307  0.1041108  4.6924 5.204e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
R-Squared      :  0.10534
      Adj. R-Squared :  0.10423
F-statistic: 22.0187 on 1 and 187 DF, p-value: 5.2035e-06
```

Net of the time trend, for every percentage point difference in people with BAs, the difference in the % of people married increases by .49 %-age points (p<.01)

Look for autocorrelation in the errors

```
> pwfdtest(plm.fd, h0 = "fd")

   Wooldridge's first-difference test for serial correlation in panels

data:  plm.fd
chisq = 54.7652, p-value = 1.358e-13
alternative hypothesis: serial correlation in differenced errors
```

Run the Wooldridge test for serial correlation within first-differenced panels. We reject the null of “no AR(1)” serial correlation. Meaning we do have serial correlation in the errors. Tricky.

Agenda

1. Lags
    - Distributed lag models
    - Lagged dependent variable
2. “Big T, Small N” panels
3. **Co-integration**
4. Granger causality

Co-integration

- If both variables have unit roots, but they may integrated to the same degree
- We can actually leave them in their levels in that case
- Because they have a *long-run* stable relationship
- Granger and Engle developed a simple test:

Step #1: Run a regression

```
> imp1 = lm(crimegrows ~ Violent.Crime.rate + year, imp.df)
> summary(imp1)

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        577.989273  78.111771   7.400 8.38e-09 ***
Violent.Crime.rate   0.026309   0.004335   6.069 5.06e-07 ***
year                -0.263464   0.038883  -6.776 5.65e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.746 on 37 degrees of freedom
Multiple R-squared:  0.7494,   Adjusted R-squared:  0.7358
F-statistic: 55.32 on 2 and 37 DF,  p-value: 7.615e-12
```

You have seen this regression before ☺

Step #2: Predict the errors

```
e1 <- imp1$resid
```

Step #3: Run a unit root test on errors

```
> urkpssTest(e1, type=c("tau"))

Title:
 KPSS Unit Root Test

  Test is of type: tau with 3 lags.

  Value of test-statistic is: 0.0559

  Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
  critical values 0.119 0.146  0.176 0.216
```

- Since 0.0559 < 0.119, then we cannot reject the null of stationarity in the residuals
- Remember that auto.arima recommended (0,0,0) for these errors already

Quick correction on the critical values

Actually, the Engle-Granger test should actually use somewhat more extreme critical values than the usual augmented DF test because of the uncertainty in the using the errors in a second stage, but they are close enough usually

Wooldridge on cointegration

Example 18.5

[Cointegration between Fertility and Personal Exemption]

In Chapters 10 and 11, we studied various models to estimate the relationship between the general fertility rate (*gfr*) and the real value of the personal tax exemption (*pe*) in the United States. The static regression results in levels and first differences are notably different. The regression in levels, with a time trend included, gives an OLS coefficient on *pe* equal to .187 (se = .035) and R^2 = .500. In first differences (without a trend), the coefficient on Δpe is −.043 (se = .028), and R^2 = .032. Although there are other reasons for these differences—such as misspecified distributed lag dynamics—the discrepancy between the levels and changes regressions suggests that we should test for cointegration. Of course, this presumes that *gfr* and *pe* are I(1) processes. This appears to be the case: the augmented DF tests, with a single lagged change and a linear time trend, each yield t statistics of about −1.47, and the estimated AR(1) coefficients are close to one.

When we obtain the residuals from the regression of *gfr* on t and *pe* and apply the augmented DF test with one lag, we obtain a t statistic on $\hat{u}_{t-1}$ of −2.43, which is nowhere near the 10% critical value, −3.50. Therefore, we must conclude that there is little evidence of cointegration between *gfr* and *pe*, even allowing for separate trends. It is very likely that the earlier regression results we obtained in levels suffer from the spurious regression problem.

The good news is that, when we used first differences and allowed for two lags—see equation (11.27)—we found an overall positive and significant long-run effect of Δpe on Δgfr.

Other tests for cointegration

There are other, newer tests for cointegration, like the Johansen test

Agenda

1. Lags
   - Distributed lag models
   - Lagged dependent variable
2. “Big T, Small N” panels
3. Co-integration
4. **Granger causality**

Granger causality I

- All it is really doing is asking: Do lags of an X have incremental predictive power beyond merely the lags of the Y variable?

- It is just an F-test for the joint significance of the lags of X, net of the lags of Y

- If we get a low p-value on the F-test of the lags of X, then we conclude that X "Granger-causes" Y

Granger causality II

- We should run lags of X on Y and then lags of Y on X too, since we are not sure of the true relationship between Y and X

- Note well: This is not a test of contemporaneous causality, but of the effect of the *lags* of X and Y on Y

The lagged model for the Granger test

```
> library(lmtest)

> grangertest(crimegrows ~ Violent.Crime.rate, order = 3, data = imp.df.ts)
Granger causality test

Model 1: crimegrows ~ Lags(crimegrows, 1:3) + Lags(Violent.Crime.rate, 1:3)
Model 2: crimegrows ~ Lags(crimegrows, 1:3)
  Res.Df Df      F  Pr(>F)
1     30
2     33 -3 3.7821 0.02056 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lags of *Violent.Crime.rate* do seem to contribute incrementally to the prediction of *crimegrows*. So *Violent.Crime.rate* does “Granger cause” *crimegrows*.

The lagged model for the Granger test

```
> grangertest(Violent.Crime.rate ~ crimegrows, order = 3, data = imp.df.ts)
Granger causality test

Model 1: Violent.Crime.rate ~ Lags(Violent.Crime.rate, 1:3) + Lags(crimegrows,
   1:3)
Model 2: Violent.Crime.rate ~ Lags(Violent.Crime.rate, 1:3)
  Res.Df Df     F Pr(>F)
1     30
2     33 -3 0.791 0.5085
```

The lags of *crimegrows* do not contribute incrementally to the prediction of *Violent.Crime.rate*. So *Violent.Crime.rate* is not "Granger caused" by *crimegrows*.

Vector autoregressive models I

- We can also run a Granger causality test another way
- First we run our model as a vector autoregressive one

What is a VAR?

Vector autoregressive model is a simultaneous equation model where:

If we have two series, y_t and z_t, a vector autoregression consists of equations that look like

$$y_t = \delta_0 + \alpha_1 y_{t-1} + \gamma_1 z_{t-1} + \alpha_2 y_{t-2} + \gamma_2 z_{t-2} + \dots \qquad [18.50]$$

and

$$z_t = \eta_0 + \beta_1 y_{t-1} + \rho_1 z_{t-1} + \beta_2 y_{t-2} + \rho_2 z_{t-2} + \dots,$$

where each equation contains an error that has zero expected value given past information on y and z.

Vector autoregressive models too

```
> var.crime <- VAR(imp.df.ts[,c("crimegrows", "Violent.Crime.rate")], p = 2)
> summary(var.crime)

VAR Estimation Results:
=========================
Estimation results for equation crimegrows:
===========================================
crimegrows = crimegrows.l1 + Violent.Crime.rate.l1 + crimegrows.l2 +
   Violent.Crime.rate.l2 + const

                      Estimate Std. Error t value Pr(>|t|)
crimegrows.l1          0.23761    0.14610   1.626  0.11339
Violent.Crime.rate.l1  0.05293    0.01950   2.715  0.01046 *
crimegrows.l2          0.46947    0.14994   3.131  0.00364 **
Violent.Crime.rate.l2 -0.04383    0.01814  -2.417  0.02136 *
const                 14.37557    8.40978   1.709  0.09677 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 3.147 on 33 degrees of freedom
Multiple R-Squared: 0.7036,   Adjusted R-squared: 0.6677
F-statistic: 19.58 on 4 and 33 DF,  p-value: 2.438e-08
```

Vector autoregressive models too

```
> var.crime <- VAR(imp.df.ts[,c("crimegrows", "Violent.Crime.rate")], p = 2)
> summary(var.crime)

VAR Estimation Results:
=========================
Estimation results for equation Violent.Crime.rate:
===================================================
Violent.Crime.rate = crimegrows.l1 + Violent.Crime.rate.l1 + crimegrows.l2 +
   Violent.Crime.rate.l2 + const

                      Estimate Std. Error t value Pr(>|t|)
crimegrows.l1          1.27065    1.06060   1.198    0.239
Violent.Crime.rate.l1  1.49655    0.14154  10.573 3.94e-12 ***
crimegrows.l2          0.05958    1.08850   0.055    0.957
Violent.Crime.rate.l2 -0.58838    0.13168  -4.468 8.76e-05 ***
const                -40.41179   61.05049  -0.662    0.513
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 22.85 on 33 degrees of freedom
Multiple R-Squared: 0.9572,   Adjusted R-squared: 0.952
F-statistic: 184.6 on 4 and 33 DF,  p-value: < 2.2e-16
```

Vector autoregressive models too

```
> causality(var.crime, cause = "crimegrows")$Granger

   Granger causality H0: crimegrows do not Granger-cause Violent.Crime.rate

data:  VAR object var.crime
F-Test = 0.912, df1 = 2, df2 = 66, p-value = 0.4067

> causality(var.crime, cause = "Violent.Crime.rate")$Granger

   Granger causality H0: Violent.Crime.rate do not Granger-cause crimegrows

data:  VAR object var.crime
F-Test = 3.7205, df1 = 2, df2 = 66, p-value = 0.02944
```

We would "accept" the null that crimegrows do not Granger-cause Violent.Crime.rate. But we would reject the null that Violent.Crime.rate do not Granger-cause crimegrows ($p=0.029$) -- I.e., Violent.Crime.rate "Granger causes" crimegrows

A spin on VAR and co-integration

How about a ECM?

What is a ECM? From Volscho & Kelly 2012

An error correction model is one :

$$\Delta Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \beta_1 \Delta X_{t-i} + \beta_2 X_{t-1} + \varepsilon_t$$

“An error correction relationship—deviations from the long-run relationship (errors) are eliminated over time through an adjustment process (error correction).”

What is a ECM? [From Volscho & Kelly 2012]

$$\Delta Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \beta_1 \Delta X_{t-i} + \beta_2 X_{t-1} + \varepsilon_t$$

"This specification allows for a test of both short- and long-run effects. The immediate short-term effect of X is captured by $\beta 1$. The error correction rate is captured by $\alpha 1$ and indicates the rate at which discrepancies between Y and X are recalibrated to their equilibrium state. Importantly, if the error correction rate is not significant, it indicates that a long-run relationship does not exist (for integrated variables this is a cointegration test). An increase in X can have an immediate impact on Y and a long-run impact that is distributed over time (dictated by the error correction rate) such that Y readjusts to the long-run equilibrium between X and Y. The total long-run impact, known as the long-run multiplier effect, is calculated by $\beta 2/\alpha 1$."

ECM ~ Traffic fatalities

```
> library(ecm)

> xeq <- xtr <- d[c('umempl')]
> model1 <- ecm(d$fatpbvmt, xeq, xtr, includeIntercept=TRUE)
> summary(model1)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0048260  0.0932949   0.052   0.9589
deltaumempl -0.0860353  0.0158717  -5.421 1.11e-06 ***
umemplLag1  -0.0001646  0.0121905  -0.014   0.9893
yLag1       -0.0284039  0.0095970  -2.960   0.0044 **
---
Residual standard error: 0.1306 on 60 degrees of freedom
Multiple R-squared:  0.4357,Adjusted R-squared:  0.4075
F-statistic: 15.44 on 3 and 60 DF,  p-value: 1.48e-07
```

ECM ~ Traffic fatalities

```
> library(ecm)
> xeq <- xtr <- d[c('umempl')]
> model1 <- ecm(d$fatpbvmt, xeq, xtr, includeIntercept=TRUE)
> summary(model1)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0048260  0.0932949   0.052   0.9589
deltaumempl -0.0860353  0.0158717  -5.421 1.11e-06 ***
umemplLag1  -0.0001646  0.0121905  -0.014   0.9893
yLag1       -0.0284039  0.0095970  -2.960   0.0044 **
---
Residual standard error: 0.1306 on 60 degrees of freedom
Multiple R-squared:  0.4357,Adjusted R-squared:  0.4075
```

In the Error Correction Model, changes in unemployment are associated with changes in traffic fatalities (-0.086***), though lags of unemployment are not critical predictors

ECM ~ Traffic fatalities

```
> library(ecm)
> xeq <- xtr <- d[c('umempl')]
> model1 <- ecm(d$fatpbvmt, xeq, xtr, includeIntercept=TRUE)
> summary(model1)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0048260  0.0932949   0.052   0.9589
deltaumempl -0.0860353  0.0158717  -5.421 1.11e-06 ***
umemplLag1  -0.0001646  0.0121905  -0.014   0.9893
yLag1       -0.0284039  0.0095970  -2.960   0.0044 **
---
Residual standard error: 0.1306 on 60 degrees of freedom
Multiple R-squared:  0.4357,Adjusted R-squared:  0.4075
```

The total long-run impact, known as the long-run multiplier effect, is calculated by -0.00016/-0.0284 = 0.00563.   Since lag_1 of unemployment is not a critical predictor, we should think if this model is fully appropriate.

Extra on our initial example

Extra on our initial example

```
> auto.arima(imp.df$crimegrows, trace=TRUE)

 ARIMA(2,1,2) with drift         : Inf
 ARIMA(0,1,0) with drift         : 212.9803
 ARIMA(1,1,0) with drift         : 208.9521
 ARIMA(0,1,1) with drift         : 208.2699
 ARIMA(0,1,0)                    : 216.4017
 ARIMA(1,1,1) with drift         : 210.7123
 ARIMA(0,1,2) with drift         : 210.7349
 ARIMA(1,1,2) with drift         : 213.3752
 ARIMA(0,1,1)                    : 212.2037

 Best model: ARIMA(0,1,1) with drift

Coefficients:
          ma1    drift
      -0.4720  -0.3099
s.e.   0.1557   0.3004

sigma^2 estimated as 12.02:  log likelihood=-100.79
AIC=207.58   AICc=208.27   BIC=212.57
```

Using auto.arima to automate unit root test.

We need first differences here.

Extra on our initial example

```
> auto.arima(imp.df$Violent.Crime.rate, trace=TRUE)

 ARIMA(2,2,2)                    : 360.4679
 ARIMA(0,2,0)                    : 355.7483
 ARIMA(1,2,0)                    : 357.3481
 ARIMA(0,2,1)                    : 357.2679
 ARIMA(1,2,1)                    : 355.1264
 ARIMA(2,2,1)                    : 360.9683
 ARIMA(1,2,2)                    : 357.4945

 Best model: ARIMA(1,2,1)

Coefficients:
         ar1      ma1
      0.5915  -0.9219
s.e.  0.1766   0.0994

sigma^2 estimated as 550.8:  log likelihood=-174.21
AIC=354.42   AICc=355.13   BIC=359.33
```

Using auto.arima to automate unit root test.

We need double differencing here.

What about 1st differences?

```
> imp.dfFD <- summarise(data.frame(imp.df),
+                       crimegrows = firstD(crimegrows), # using firstD functon
   from QMSS package
+                       Violent.Crime.rate = firstD(Violent.Crime.rate),
+                       year= firstD(year))
>
> imp2FD <- update(imp1, data = imp.dfFD)
> summary(imp2FD)

Call:
lm(formula = crimegrows ~ Violent.Crime.rate + year, data = imp.dfFD)

Coefficients: (1 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.22763    0.66361  -0.343    0.734
Violent.Crime.rate  0.01385    0.02301   0.602    0.551
year                     NA         NA      NA       NA

Residual standard error: 4.143 on 37 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.009692,    Adjusted R-squared:  -0.01707
F-statistic: 0.3621 on 1 and 37 DF,  p-value: 0.551
```

Or 2nd diffs?

```
> imp.dfSD <- summarise(data.frame(imp.dfFD),
+                       crimegrows = firstD(crimegrows), # using firstD functon
   from QMSS package
+                       Violent.Crime.rate = firstD(Violent.Crime.rate),
+                       year= firstD(year))
>
> imp2SD <- update(imp1, data = imp.dfSD)
> summary(imp2SD)

Call:
lm(formula = crimegrows ~ Violent.Crime.rate + year, data = imp.dfSD)

Coefficients: (1 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.10061    1.21408  -0.083    0.934
Violent.Crime.rate -0.03384    0.04783  -0.708    0.484
year                     NA         NA      NA       NA

Residual standard error: 7.476 on 36 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.01372, Adjusted R-squared:  -0.01368
F-statistic: 0.5007 on 1 and 36 DF,  p-value: 0.4838
```