2019 Essex Summer School
3K: Dynamics and Heterogeneity

Robert W. Walker, Ph. D.
Associate Professor of Quantitative Methods
Atkinson Graduate School of Management
Willamette University
Salem, Oregon USA
rwalker@willamette.edu

August 6, 2019

Three Standard Time-Serial Structures [ARIMA]

AutoRegressive Integrated Moving Average (ARIMA) structures characterize most time series of interest (virtually all with the integration of their seasonal counterparts). In general, we write

- Autoregression [AR(p)]:

$$e_t = \rho_1 e_{t-1} + \rho_2 e_{t-2} + \cdots + \rho_p e_{t-p} + \nu_t$$

- Moving Average [MA(q)]:

$$e_t = \nu_t + \theta_1 \nu_{t-1} + \theta_2 \nu_{t-2} + \cdots + \theta_p \nu_{t-q}$$

- Autoregression and Moving Average [ARMA(p, q)]:

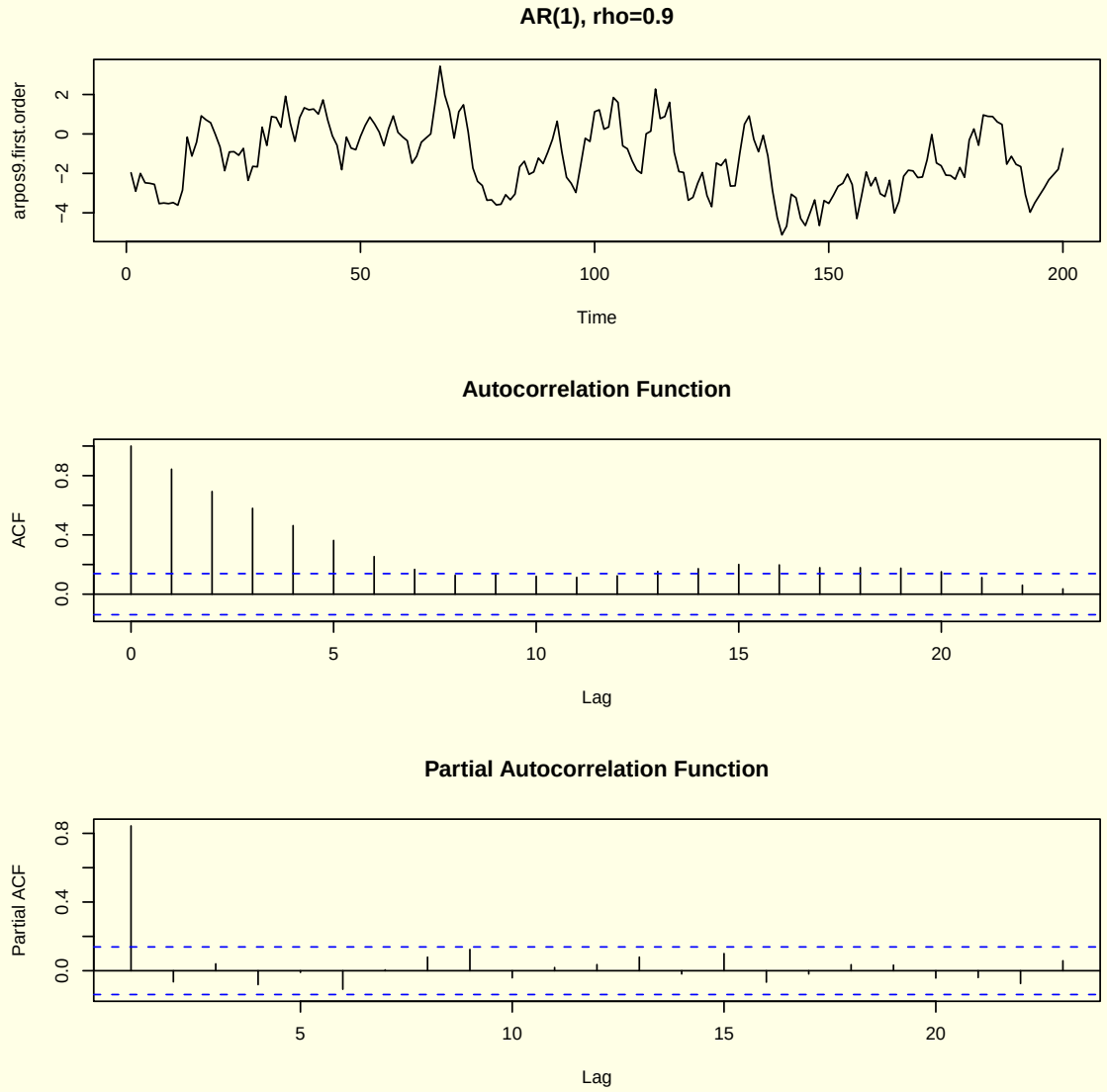
$$e_t = \rho_1 e_{t-1} + \rho_2 e_{t-2} + \cdots + \rho_p e_{t-p} + \nu_t + \theta_1 \nu_{t-1} + \theta_2 \nu_{t-2} + \cdots + \theta_q \nu_{t-q}$$

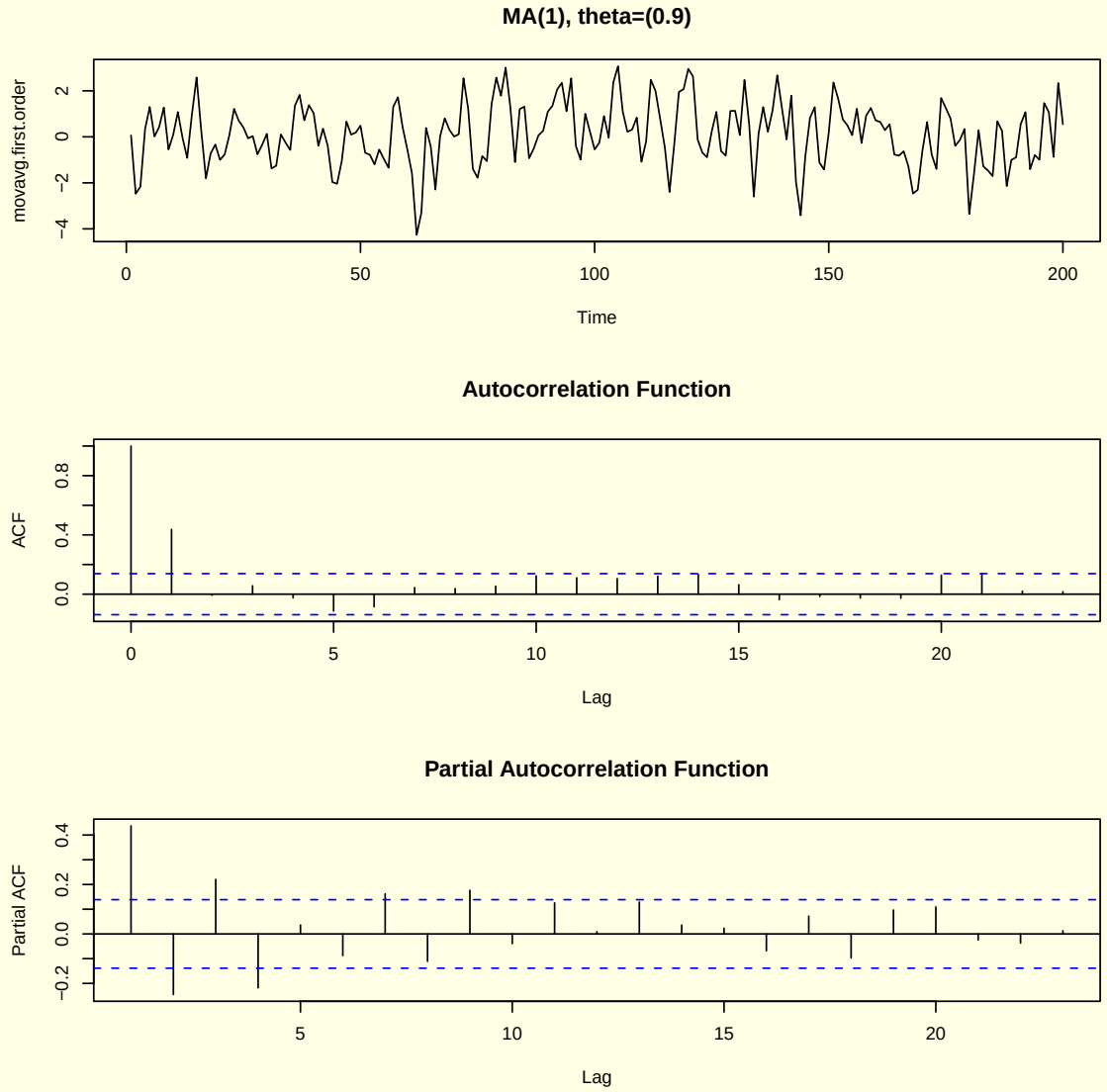
Before, I mentioned two relevant autocorrelations and discussed them briefly:

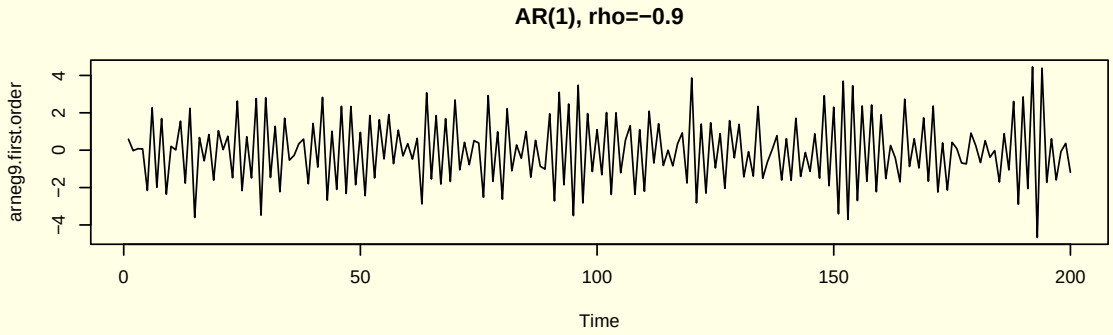
1. Autocorrelation: $\rho_s = \frac{\sum_{t=s+1}^{T}(y_t - \overline{y})(y_{t-s} - \overline{y})}{\sum_{t=1}^{T}(y_t - \overline{y})^2}$

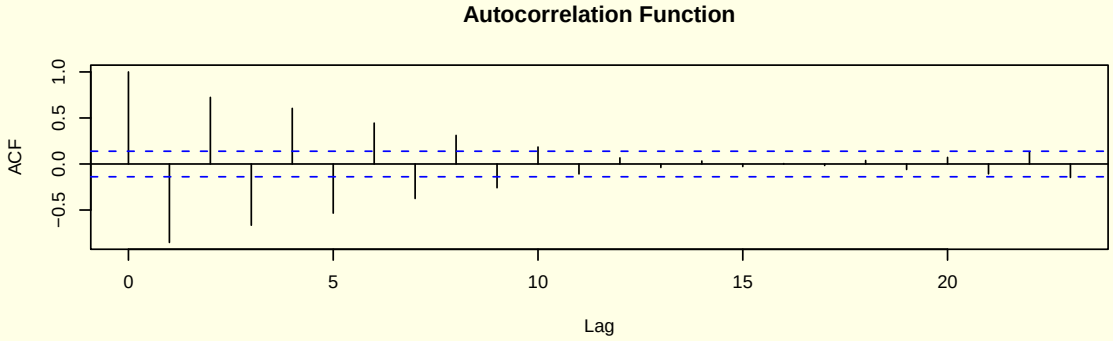
2. Partial Autocorrelation $\phi_s = \frac{\rho_s - \sum_{j=1}^{s-1} \phi_{s-1,j}\rho_{s-j}}{1 - \sum_{j=1}^{s-1} \phi_{s-1,j}\rho_j}$

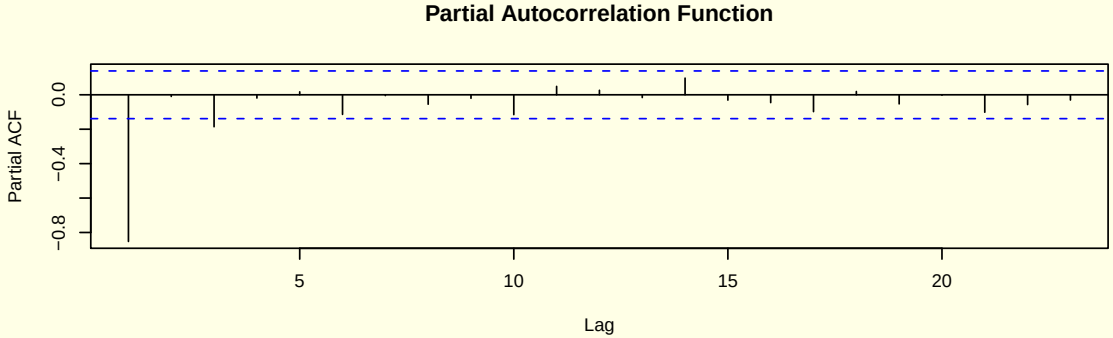
In ARIMA modeling, these are two critical components as each process has a characteristic signature. An autoregressive process typically exhibits geometric decay in the autocorrelation function and spikes in the partial; moving average processes exhibit the reverse. Nonstationary series decay very slowly (the I in ARIMA).

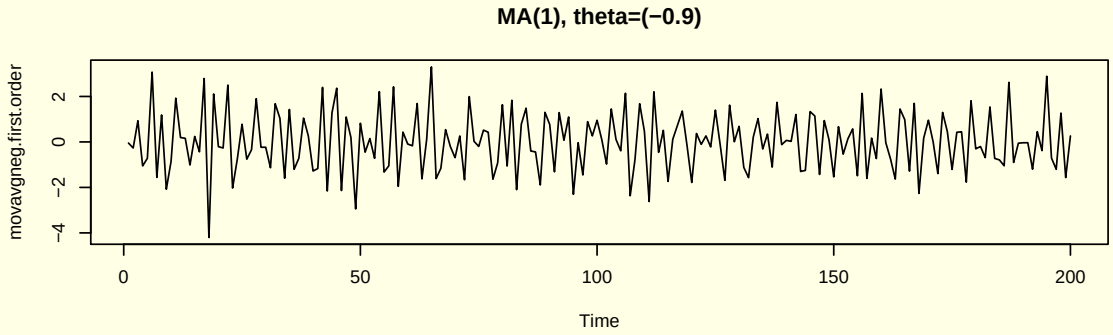

AR(1), rho=0.9
arpos9.first.order
−4
−2
0
2
0
50
100
150
200
Time
Autocorrelation Function
ACF
0.0
0.4
0.8
0
5
10
15
20
Lag
Partial Autocorrelation Function
Partial ACF
0.0
0.4
0.8
5
10
15
20
Lag

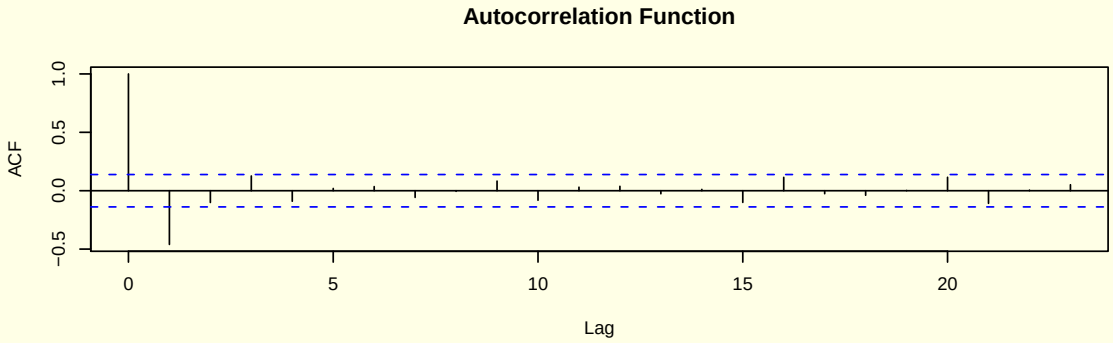

MA(1), theta=(0.9)
movavg.first.order
−4
−2
0
2
0
50
100
150
200
Time
Autocorrelation Function
ACF
0.0
0.4
0.8
0
5
10
15
20
Lag
Partial Autocorrelation Function
Partial ACF
−0.2
0.0
0.2
0.4
5
10
15
20
Lag

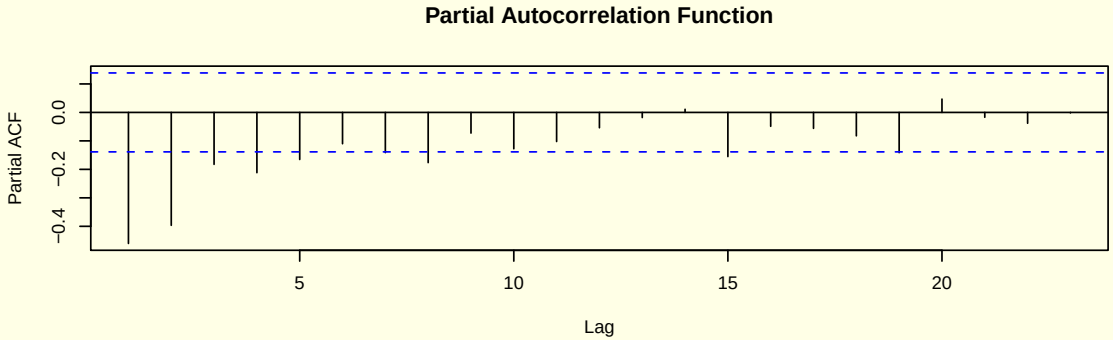

AR(1), rho=−0.9
arneg9.first.order
−4
−2
0
2
4
0
50
100
150
200
Time

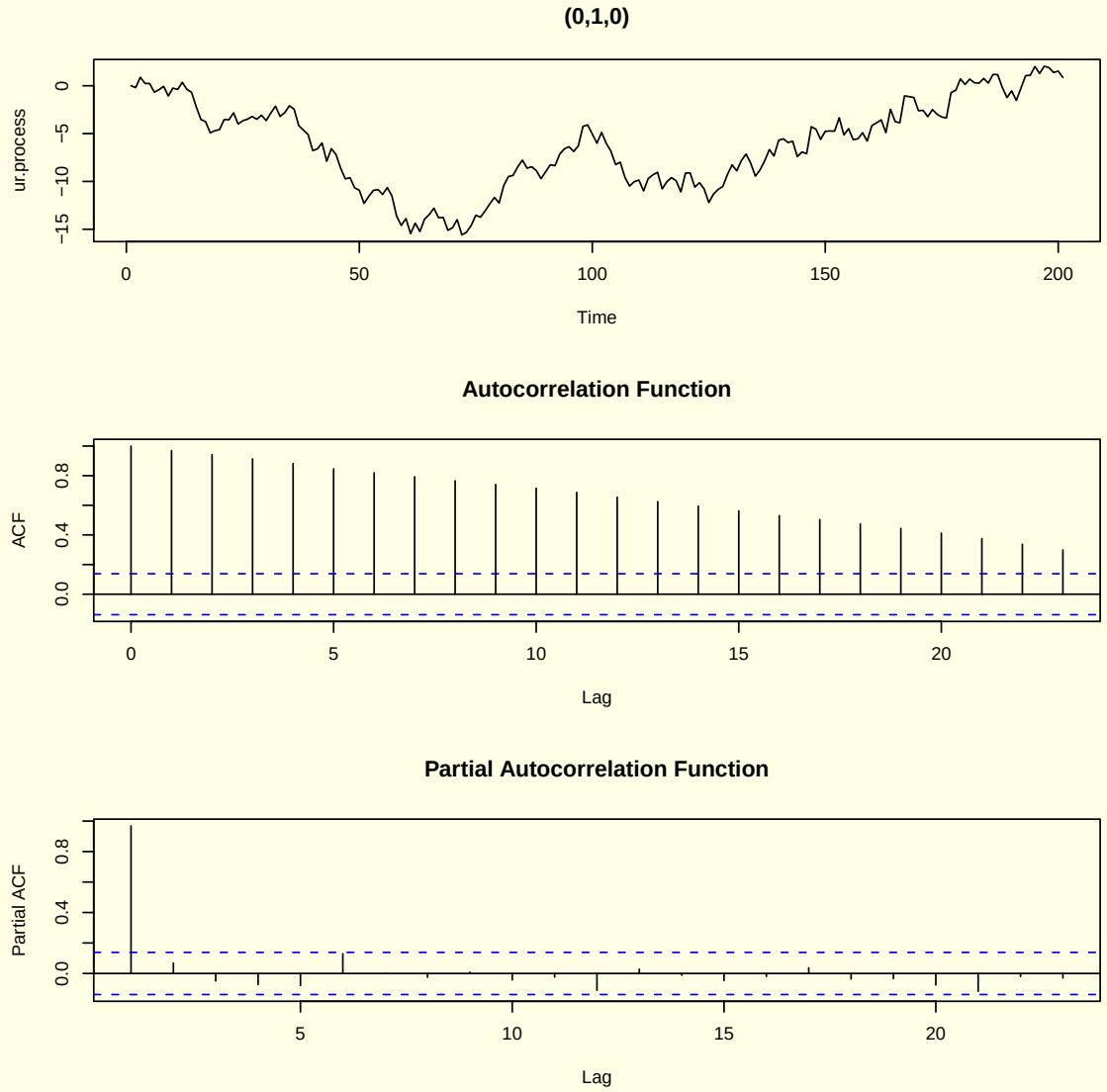

Autocorrelation Function
ACF
−0.5
0.0
0.5
1.0
0
5
10
15
20
Lag


Partial Autocorrelation Function
Partial ACF
−0.8
−0.4
0.0
5
10
15
20
Lag


MA(1), theta=(−0.9)
movavgneg.first.order
−4
−2
0
2
0
50
100
150
200
Time


Autocorrelation Function
ACF
−0.5
0.0
0.5
1.0
0
5
10
15
20
Lag


Partial Autocorrelation Function
Partial ACF
−0.4
−0.2
0.0
5
10
15
20
Lag


(0,1,0)
ur.process
0
−5
−10
−15
0
50
100
150
200
Time
Autocorrelation Function
ACF
0.0
0.4
0.8
0
5
10
15
20
Lag
Partial Autocorrelation Function
Partial ACF
0.0
0.4
0.8
5
10
15
20
Lag

Stata

Though these plots were generated in R, we could do the same thing in Stata. For a quick summary with a little graphic, have a look at the `corrgram`. For (pretty) plots, Stata has two commands to recreate this, `ac` and `pac`. The former generates the autocorrelations while the latter creates the partial autocorrelations. We will have a go at this in the lab.

TSCS and Time Series

- Common structure restrictions may be difficult to deal with and limit our ability to gain much from combining individual time series.
- Most will be pretty simple structures.
- Mixed orders of integration present special problems.

Diagnosing Serial Correlation Individually

If we can reject a range of pathologies, we can justify inference rationally?

- First question is the integrity of the estimand; does the conditional mean make sense?

- Unit root tests come in a host of forms with nulls of a unit root and nulls of stationarity. The processes have different implications. Unfortunately, in TSCS/CSTS settings, tests are pretty unreliable. That said,

  ⋆ Levin and Lin: `levinlin` with $H_0 : I(1)$.
  ⋆ Im, Pesaran, and Shin: `ipshin` with $H_0 : I(1)$.
  ⋆ KPSS: `kpss` with $H_0 : I(0)$.
  ⋆ Fisher: `xtfisher` works with unbalanced panels
  ⋆ Simple `xtreg` with lagged y, if $\beta_{y_{t-1}} \approx 1$ then there is a worry.

- Given this:
  - ⋆ Plots (Every structure has different theoretical ACF/PACF)
  - ⋆ Durbin-Watson d and Durbin's h with endogenous variables
  - ⋆ Dickey-Fuller tests and many others. $\Delta y_t = \rho y_{t-1} + \theta_L \Delta y_{t-L} + \lambda_t + u_t$
  - ⋆ Breusch-Godfrey test and the like (Fit regression, isolate residuals, regress residual on X and lags of residual, $nR^2 \sim \chi^2_p$).

- The above alongside:
  (1) is the temporal process common or distinct? and
  (2) if distinct, how and why?

Panel Unit Root Testing in Stata

As of Stata 11, a battery of panel unit-root tests have emerged. There are many and they operate under differing sets of assumptions.

- Levin-Lin-Chu (`xtunitroot llc`): trend nocons (unit specific) demean (within transform) lags. Under (crucial) cross-sectional independence, the test is an advancement on the generic Dickey-Fuller theory that allows the lag lengths to vary by cross-sections. The test relies on specifying a kernel (beyond our purposes) and a lag length (upper bound). The test statistic has a standard normal basis with asymptotics in $\frac{\sqrt{N_T}}{T}$ (T grows faster than N). The test is of either all series containing unit roots (H_0) or all stationary; this is a limitation. It is recommended for moderate to large T and N.

1. Perform separate ADF regressions:

$$\Delta y_{it} = \rho_i \Delta y_{i,t-1} + \sum_{L=1}^{p_i} \theta_{iL} \Delta y_{i,t=L} + \alpha_{mi} d_{mt} + \epsilon_{it}$$

with d_{mt} as the vector of deterministic variables (none, drift, drift and trend). Select a max L and use t on $\hat{\theta}_{iL}$ to attempt to simplify. Then use $\Delta y_{it} = \Delta y_{i,t-L}$ and d_{mt} for residuals

- Harris-Tzavalis (`xtunitroot ht`): trend nocons (unit specific) demean (within transform) altt (small sample adjust) Similar to the previous, they show that $T \to \infty$ faster than N (rather than T fixed) leads to size distortions.

- Breitung (`xtunitroot breitung`): trend nocons (unit specific) demean (within transform) robust (CSD) lags.
  Similar to LLC with a common statistic across all i.

- Im, Pesaran, Shin (`xtunitroot ips`): trend demean (within transform) lags. They free ρ to be ρ_i and average individual unit root statistics. The null is that all contain unit roots while the alternative specifies at least some to be stationary. The test relies on sequential asymptotics (first T, then N). Better in small samples than LLC, but note the differences in the alternatives.

- Fisher type tests (`xtunitroot fisher`): dfuller pperron demean lags.

- Hadri (LM) (`xtunitroot hadri`): trend demean robust

All but the last are null hypothesis unit-root tests. Most assume balance but the fisher and IPS versions can work for unbalanced panels.

Day VI: Review, Summary, and to Missing Data

Stationarity Issues

- Essence of stationarity is threefold: means, variances, and crosses are not time-dependent.
- There is a quite famous spurious regressions result in econometrics that owes to the statistician Yule in 1926.
- Basically, the regression of $I(1)$ series on one another has non-α rejection rates.
- Applied to panels, a mix of orders of integration will give t statistics non-t properties.
- In the end, I suspect the best advice is to partition data on the basis of likely orders of integration and proceed from there.

ADL/Canonical models

We can consider some very basic time series models.

- Koyck/Geometric decay:
  short run and long-run effects are parametrically identified (given $\mathcal{M}$).

- Almon (more arbitrary decay):

$$y_{it} = \sum_{t_A=0}^{T_F} \rho_{t_A} x_{t-t_A} + \epsilon_t$$

  with coefficients that are ordinates of some general polynomial of degree $T_F >> q$. The $\rho_{t_A} = \sum_{k=0}^{T_F} \gamma_k t^k$.

- Prais-Winston, etc. are basically FGLS implementations of AR(1).

Prais-Winsten/Cochrane-Orcutt

$$y_{it} = X_{it}\beta + \epsilon_{it}$$

where

$$\epsilon_{it} = \rho\epsilon_{i,t-1} + \nu_{it}$$

and $\nu_{it} \sim N(0, \sigma_\nu^2)$ with stationarity forcing $|\rho| < 1$. We will use iterated FGLS. First, estimate the regression recalling our unbiasedness condition. Then regress $\hat{\epsilon}_{it}$ on $\hat{\epsilon}_{i,t-1}$. Rinse and repeat until ρ doesn't change. The transformation applied to the first observation is distinct, you can look this up.... In general, the transformed regression is:

$$y_{it} - \rho y_{i,t-1} = \alpha(1-\rho) + \beta(X_{it} - \rho X_{i,t-1}) + \nu_{it}$$

with ν white noise.

Beck

- Static model: Instantaneous impact.

$$y_{i,t} = X_{i,t}\beta + \nu_{i,t}$$

- Finite distributed lag: lags of x finite horizon impact (defined by lags).

$$y_{i,t} = X_{i,t}\beta + \sum_{k=1}^{K} X_{i,t-k}\beta_k + \nu_{i,t}$$

- AR(1): Errors decay geometrically, X instantaneous. (Suppose unmeasured x and think this through).

$$y_{i,t} = X_{i,t}\beta + \nu_{i,t} + \theta\epsilon_{i,t-1}$$

- Lagged dependent variable: lags of y [common geometric decay]

$$y_{i,t} = X_{i,t}\beta + \phi y_{i,t-1} + \nu_{i,t}$$

- ADL: current and lagged x and lagged y.

$$y_{i,t} = X_{i,t}\beta + X_{i,t-1}\gamma + \phi y_{i,t-1} + \epsilon_{i,t}$$

- Panel versions of transfer function models from Box and Jenkins time series. (each x has an impact and decay function)

Interpretation of dynamic models

- Do it.
- Whitten and Williams `dynsim` uses `Clarify`[1] to do this.
- Their paper is "But Wait, There's More! Maximizing Substantive Inferences from TSCS Models". Easy to find on the web and on the website.

[1] If you do not know what Clarify is, please ask: estimate, set, simulate.

Bringing Time and Space Together

Wilson and Butler

- Survey of papers using TSCS data and methods(?)
- Vast majority do nothing about space or time.
- Does it matter?

  Table 3
  Table 4

- What do we do? Raise the bar for positive findings and look at multiple models trying to tease out the role of particular assumptions as necessary and/or sufficient for results.

More on xtpcse

Holding on to data

- `preserve`
- `restore`

Testing the Null Hypothesis of No Random Effects

```
. xttest0

Breusch and Pagan Lagrangian multiplier test for random effects:

        growth[country,t] = Xb + u[country] + e[country,t]

        Estimated results:
                         |       Var     sd = sqrt(Var)
                ---------+-----------------------------
                  growth |   6.041246       2.457895
                       e |   4.147091       2.036441
                       u |   .0473477       .2175953

        Test:   Var(u) = 0
                                 chi2(1) =     4.39
```

```
Prob > chi2 =     0.0361
```

xttest

```
. xttest1

Tests for the error component model:

        growth[country,t] = Xb + u[country] + v[country,t]
           v[country,t] = rho v[country,(t-1)] + e[country,t]

        Estimated results:
                                 Var     sd = sqrt(Var)
                ---------+-----------------------------
                  growth |   6.041246       2.457895
                       e |   4.037869      2.0094449
                       u |     .13335      .36517121

        Tests:
           Random Effects, Two Sided:
           LM(Var(u)=0)       =    1.00   Pr>chi2(1) =  0.3174
           ALM(Var(u)=0)      =    0.54   Pr>chi2(1) =  0.4610

           Random Effects, One Sided:
           LM(Var(u)=0)       =    1.00   Pr>N(0,1)  =  0.1587
           ALM(Var(u)=0)      =    0.74   Pr>N(0,1)  =  0.2305

           Serial Correlation:
           LM(rho=0)          =    0.74   Pr>chi2(1) =  0.3906
```

```
            ALM(rho=0)          =    0.28   Pr>chi2(1) =  0.5961

            Joint Test:
            LM(Var(u)=0,rho=0) =    1.28   Pr>chi2(2) =  0.5271
```

```
* We cannot reject the null hypothesis of no variation in the random effects.
Also no evidence of serial correlation.
Remember, with the lagged endogenous variable on the right hand side,
the random effects are included if they are there.
```

`xttest1`

1. LM test for random effects, assuming no serial correlation
2. Adjusted LM test for random effects, which works even under serial correlation
3. One-sided version of the LM test for random effects
4. One-sided version of the adjusted LM test for random effects
5. LM joint test for random effects and serial correlation
6. LM test for first-order serial correlation, assuming no random effects
7. Adjusted test for first-order serial correlation, which works even under random effects

`xtgls`

- `corr`: t structure ([ar] or [ps]ar) is ρ common or not.
- `panels`: i structure (iid, [h]eteroscedastic, [c]orrelated (and [h]))
- `rhotype`: regress (regression using lags), dw - Durbin-Watson, freg (forward regression uses leads), nagar, theil, tscorr
- `igls` (iterate or two-step)
- `force` for unbalanced.

`xttest2` and `xttest3`

After `fe` or `xtgls`, we have two tests pre-programmed.

1. We have a test of independence (within) in `xttest2`
2. We have a test of homoscedasticity (within) in `xttest3`

`xtserial`

Wooldridge presents a test for serial correlation.

`xtcsd`

How do we test for cross-sectional dependence?

- Generally used for small T and large N settings.
- Three methods: `xtcsd, pesaran friedman frees`
- This is the panel correction in PCSE

xtscc

Driscoll and Kraay (1998) describe a robust covariance matrix estimator for pooled and fixed effects regression models that contain a large time dimension. The approach is robust to heteroscedasticity, autocorrelation, and spatial correlation.

We're Here for Fancy Estimators, Why is Everything OLS?

There are limitation imposed by what people have programmed in terms of regression diagnostics. However, if we can fit the same model by OLS, we can use standard regression diagnostics post-estimation to avoid calculating the diagnostics by hand. Many diagnostics are pre-programmed.

OLS Diagnostics

- We could also use other standard diagnostics in the OLS framework. If you are going to intensively use Stata, books like Statistics with Stata are quite useful.

  `estat ovtest, [rhs]` will give us Ramsey's RESET test. The option gives us RHS variables, otherwise we just use fitted values. The default is a Wald test applied to the regression

$$y_{it} = X_{it}\beta + \hat{y}^2\gamma_1 + \hat{y}^3\gamma_2 + \hat{y}^4\gamma_3 + \epsilon_{it}$$

  and with option `rhs` the powers are applied to the right-hand side variables.

  `predict ... , dfits` and `dfbeta`: We also have the various `dffits` and `dfbeta` statistics for use in diagnosing leverage. The dfit is the studentized residual multiplied by the square root of h_j over $(1 - h_j)$;

basically a scaled measure of the difference between in-sample and out-of-sample predictions. The `dfit` is obtained as a post-regression prediction using predict. Define `dfbeta` as:

$$DFBETA_j = \frac{r_j v_j}{\sqrt{v^2(1-h_j)}}$$

where h is the j^{th} item in $\mathbf{P}$, r_j is the studentized residual, v_j are the residuals from a regression not containing the regressor in question, and v^2 is their sum of squares. Suggested cutoffs are $2\sqrt{\frac{k}{N}}$ for dfit and $\frac{2}{\sqrt{N}}$ for dfbeta. There is also the Cook's distance (`cooksd`) and Welsch distance (`welsch`).

`estat hettest [varlist] [, rhs [normal | iid | fstat] mtest[(spec)]]` gives us a variety of tests for heteroscedasticity. The `rhs` option gives structure from covariates. `mtest` is important because we are doing multiple testing

(often).

`estat vif` gives us some collinearity diagnostics. The statistic is essentially $\frac{1}{1-R^2_{(-k)}}$.

`estat imtest [, preserve white]` where the default is Cameron-Trivedi, we can request White's version, and preserve maintains the original data (saves time often). As a general misspecification test, the Information Matrix test is shown by Hall (1987) to decompose into heteroscedasticity, skewness, and kurtosis of residuals and has some suboptimal properties.

Plots

- avplot: added-variable plot
- avplots: all added-variable plots in one image
- cprplot: component-plus-residual plot
- lvr2plot: leverage-versus-squared-residual plot
- rvfplot: residual-versus-fitted plot
- rvpplot: residual-versus-predictor plot

Panel Data Models: SUR

The seemingly unrelated regressions model owes to early work by Zellner (1962).

$$
\begin{aligned}
y_1 &= \mathbf{X_1}\beta_1 + \epsilon_1 && (1)\\
y_2 &= \mathbf{X_2}\beta_2 + \epsilon_2 && (2)\\
\vdots &= \vdots && (3)\\
y_G &= \mathbf{X_G}\beta_G + \epsilon_G && (4)
\end{aligned}
$$

The basic idea is to estimate the above equations and the covariance matrix of disturbances across equations. They are called seemingly unrelated because the structure is quite general.

Implementation

```
sureg (depvar1 varlist1) (depvar2 varlist2) ...
   (depvarN varlistN) [if] [in] [weight]
```

There is also a general suite of seemingly unrelated estimation in `suest`.

An Example

Need to transform data structure to wide or use suest

```
regress caplab lagcaplab $RHS if cc==4
estimates store cc4
regress caplab lagcaplab $RHS if cc==6
estimates store cc6
suest cc4 cc6
test [cc6_mean=cc4_mean], cons
```

```
. suest

Simultaneous results for cc6, cc4

                                                  Number of obs   =         54

------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
cc6_mean     |
   lagcaplab |   .8469003   .2509837     3.37   0.001     .3549814    1.338819
        unem |  -.0297673    .018588    -1.60   0.109    -.0661992    .0066646
    depratio |  -.0450716   .0573967    -0.79   0.432     -.157567    .0674239
        left |   -.004146   .0015278    -2.71   0.007    -.0071405   -.0011516
       trade |   -.007032   .0054463    -1.29   0.197    -.0177064    .0036425
      caplib |   .0228315   .0223524     1.02   0.307    -.0209784    .0666414
       _cons |   2.131437   2.102184     1.01   0.311    -1.988769    6.251643
-------------+----------------------------------------------------------------
cc6_lnvar    |
       _cons |  -4.786597    .191951   -24.94   0.000    -5.162814    -4.41038
-------------+----------------------------------------------------------------
cc4_mean     |
   lagcaplab |   .0186575   .1085452     0.17   0.864    -.1940872    .2314023
        unem |   -.045907   .0162468    -2.83   0.005    -.0777501   -.0140639
    depratio |   .0372976   .0237657     1.57   0.117    -.0092823    .0838775
```

```
        left |  (dropped)
       trade |  -.0257866   .0096947    -2.66   0.008    -.0447878   -.0067853
      caplib |  -.0883456   .0304022    -2.91   0.004    -.1479328   -.0287585
       _cons |   3.162133   1.504143     2.10   0.036     .2140676    6.110199
-------------+----------------------------------------------------------------
cc4_lnvar    |
       _cons |  -4.739128   .1701558   -27.85   0.000    -5.072627   -4.405629
------------------------------------------------------------------------------
```

```
. test [cc6_mean=cc4_mean], cons

 ( 1)  [cc6_mean]lagcaplab - [cc4_mean]lagcaplab = 0
 ( 2)  [cc6_mean]unem - [cc4_mean]unem = 0
 ( 3)  [cc6_mean]depratio - [cc4_mean]depratio = 0
 ( 4)  [cc6_mean]left - [cc4_mean]left = 0
 ( 5)  [cc6_mean]trade - [cc4_mean]trade = 0
 ( 6)  [cc6_mean]caplib - [cc4_mean]caplib = 0
 ( 7)  [cc6_mean]_cons - [cc4_mean]_cons = 0

           chi2(  7) =   40.79
         Prob > chi2 =    0.0000
* Coefficients are not all the same.
```

Implementation

- `xtregar`: , `re` and `fe` options

  Fit a first order autoregressive structure to TSCS data.
  Defaults to an iterative estimator but `twostep` is available.
  `lbi` gives a test of the hypothesis that ρ is zero. (not a default)

- `xtabond`

  `estat abond` gives a test for autocorrelation
  `estat sargan` gives the overidentifying restrictions test

- `xtlsdvc y x, initial(ah or ab or bb) vcov(1000 bs iter)` will handle unbalanced
  Bias-corrected least-squares dummy variable (LSDV) estimators for the standard autoregressive panel-data model using the bias approximations in Bruno (2005a) for unbalanced panels

- `xtivreg`

- `xtdpd` fits Arellano-Bond and Arellano-Bover/Blundell-Bond

  `estat abond` gives a test for autocorrelation
  `estat sargan` gives the overidentifying restrictions test (Rejection implies failure of assumptions)

Day VII: Panel GLM: Unit Heterogeneity

Dirty Pool

1. Green, Kim, Yoon point out that the addition of dyad fixed effects undercut much of what we think we know about the causes of dyadic trade and war.

2. Oneal and Russett expand the data, argue that the approach is not very sound (especially for the conflicts), and that the critique is overstated.

3. Beck and Katz make a compelling case for problems with the application of fixed effects estimators to these binary circumstances, especially with rare events.

4. King provides a nice summary of the issues and brings much of the debate to some central points.

A Brief MLE Background using the Binary Example

- Data type: Binary data
- Densities: Mostly symmetric and unimodal [exception: cloglog]
- Interpretation: Nonlinear models require the specification of the entire prediction vector. Standard errors are not nuisance parameters.
- Testing: Z-scores and Wald tests [bad small sample properties]. Likelihood Ratio.

Data Types

- Survey responses: {yes, no}
- Choices: {Capital controls, Free Capital Movements}
- Votes in a Two-Party System {Democrat, Republican}
- A Host of Others...

Typical Interpretation:
Some unobserved continuous random variable crosses a threshold.

Binary Choice Models

The generic distribution for a binary choice situation is the Bernoulli distribution. A Bernoulli trial requires two first principles based on two alternatives:

1. Mutual exclusivity $\Rightarrow Pr(y = 1|y = 0) = 0$ or $Pr(y = 0|y = 1) = 0$

2. Exhaustion $\Rightarrow Pr(y = 1) = 1 - Pr(y = 0)$

If we define

$$\pi = Pr(y = 1) \tag{5}$$

Relying on (5) and Exhaustion, we can define the probability mass function for a Bernoulli random variable as:

$$Y_i \sim f_{Bernoulli}(y_i|\pi) = \begin{cases} \pi^{y_i}(1-\pi)^{1-y_i} & \text{for y=0,1} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Binary Choice Models, II

Let u_i^* be some unobserved latent utility for an action y_i taken by actor i. Though we cannot observe u_i^*, expected utility theory requires that taking an action generate nonnegative utility. As a result, we should only observe $y_i = 1$ if $u_i^* \geq 0$. In compact notation with $1 = T$ and $0 = F$, $y_i = I[u_i^* \geq 0]$. To derive the maximum likelihood estimator is fairly simple, we simply need the product of y_i multiplied by the action probability (π_i) associated with y_i. Thus,

$$\mathcal{L}(\pi|\vec{y}) = \prod_{i=1}^{N} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

Of course, this model is not very interesting, because we have only have data on $\vec{y}$ and the MLE of π is simply $\frac{1}{N}\sum_{i=1}^{N} y_i$. What we really want is a conditional mean function for u_i^* and an assumption about the distribution of the latent errors. Let's consider some candidates.

Specifying a Conditional Mean Function

Let

$$u_i^* = X_i\beta + \epsilon_i$$

The conditional expectation of u_i^* is a linear additive function of X_i and unobserved parameters β. The way to concoct probabilities from this setup is to engage in a bit of transformation. So, let's substitute,

$$u_i^* = X_i\beta \geq -\epsilon_i$$

If the distribution is symmetric [about zero as are logistics and normals] we can engage in some trickery,

$$\begin{aligned} Pr(y = 1) &= F(-X\beta) \quad (7) \\ &= \Lambda(-X\beta) \quad (8) \end{aligned}$$

This allows us to write the relevant likelihoods as follows:

$$\mathcal{L}_N(\beta|y_i, X_i) = \prod_{i=1}^{N} \Phi(-X_i\beta)^{y_i}\Phi(X_i\beta)^{1-y_i} \quad (9)$$

$$\mathcal{L}_L(\beta|y_i, X_i) = \prod_{i=1}^{N} \Lambda(-X_i\beta)^{y_i}\Lambda(X_i\beta)^{1-y_i} \quad (10)$$

and to write the logs of the likelihood functions as

$$\ln \mathcal{L}_N(\beta|y_i, X_i) = \sum_{i=1}^{N} y_i \ln \Phi(-X_i\beta) + (1-y_i) \ln \Phi(X_i\beta) \quad (11)$$

$$\ln \mathcal{L}_L(\beta|y_i, X_i) = \sum_{i=1}^{N} y_i \ln \Lambda(-X_i\beta) + (1-y_i) \ln \Lambda(X_i\beta) \quad (12)$$

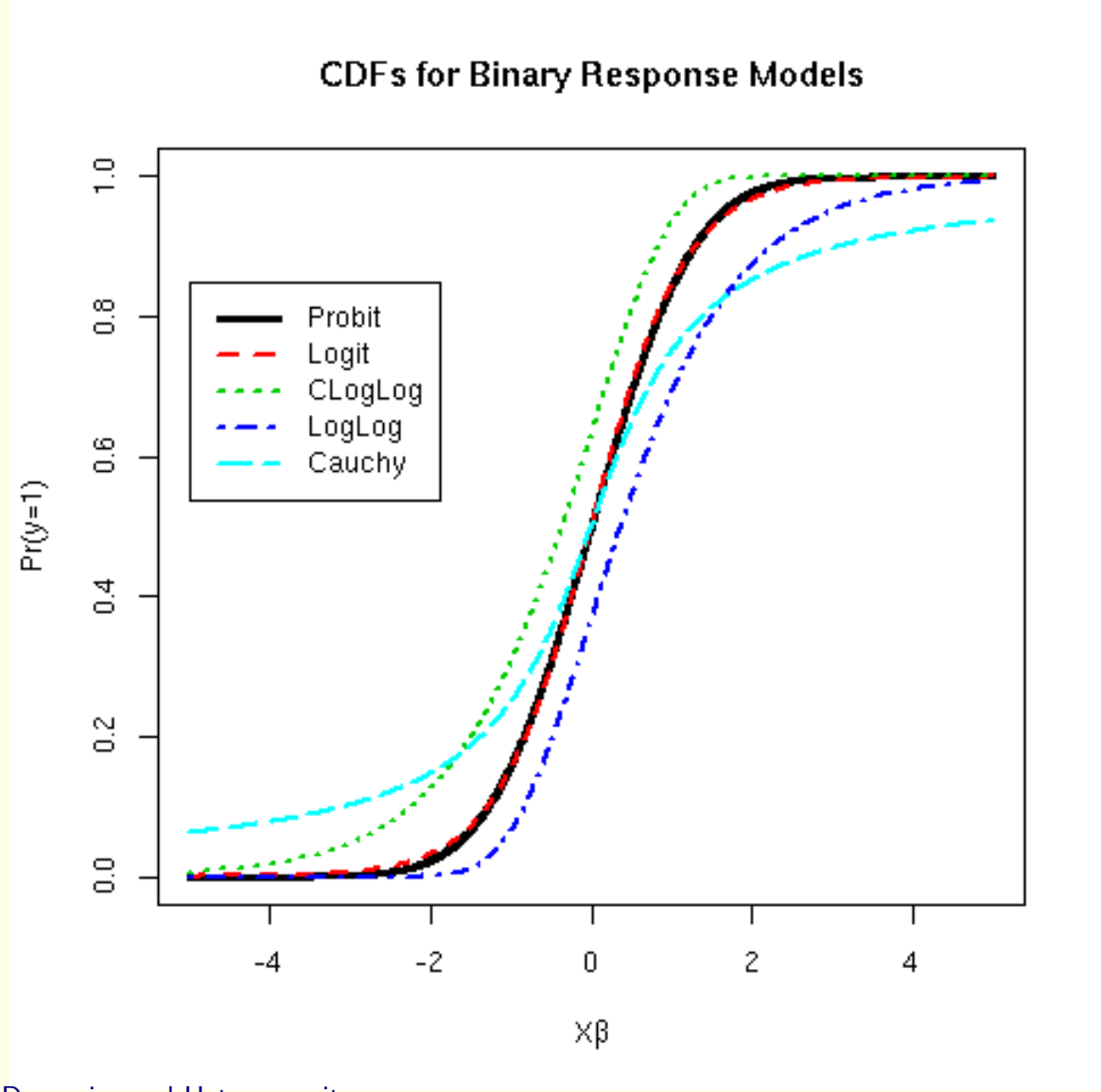
Or, for arbitrary c.d.f. F,

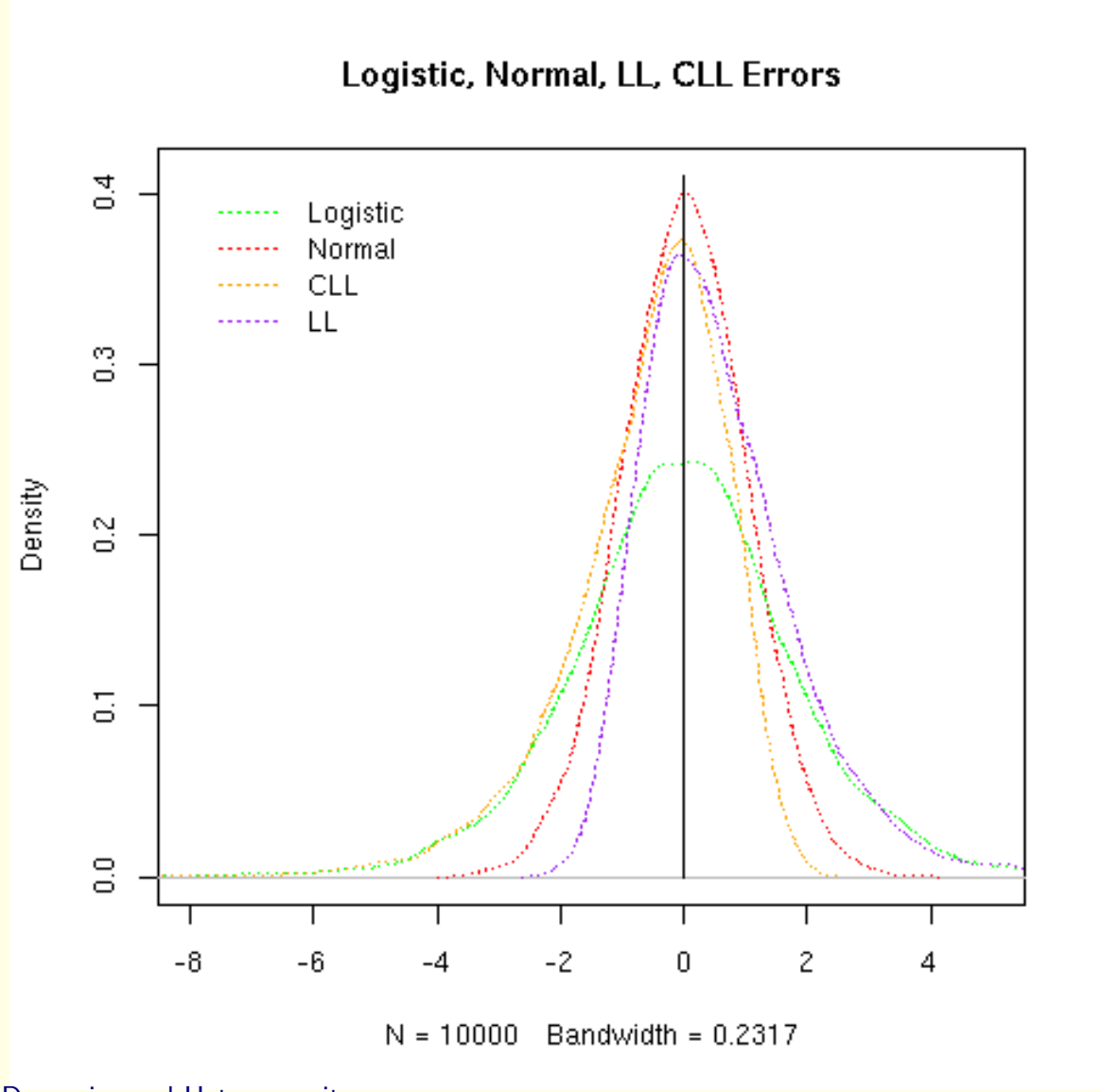
$$\ln \mathcal{L} = \sum_{i=1}^{N} (1 - y_i) \ln F(X\beta) + y \ln(1 - F(X\beta))$$

This gives us everything we need for logits, probits, and a host of other Bernoulli-based MLE's.

Some cumulative distribution functions:

Normal [S]	Φ which does not have a closed form integral
Logistic [S]	$\Lambda = \frac{e^{X_i\beta}}{1+e^{X_i\beta}}$
Cauchy [S]	$\frac{1}{\pi} \arctan(X\beta) + \frac{1}{2}$
Complementary log-log:	$1 - e^{-e^{X_i\beta}}$
Log-log:	$e^{-e^{-X_i\beta}}$


CDFs for Binary Response Models
Pr(y=1)
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Xβ
Probit
Logit
CLogLog
LogLog
Cauchy


Logistic, Normal, LL, CLL Errors
Logistic
Normal
CLL
LL
Density
0.0
0.1
0.2
0.3
0.4
-8
-6
-4
-2
0
2
4
N = 10000 Bandwidth = 0.2317

Identification of Binary Choice Models

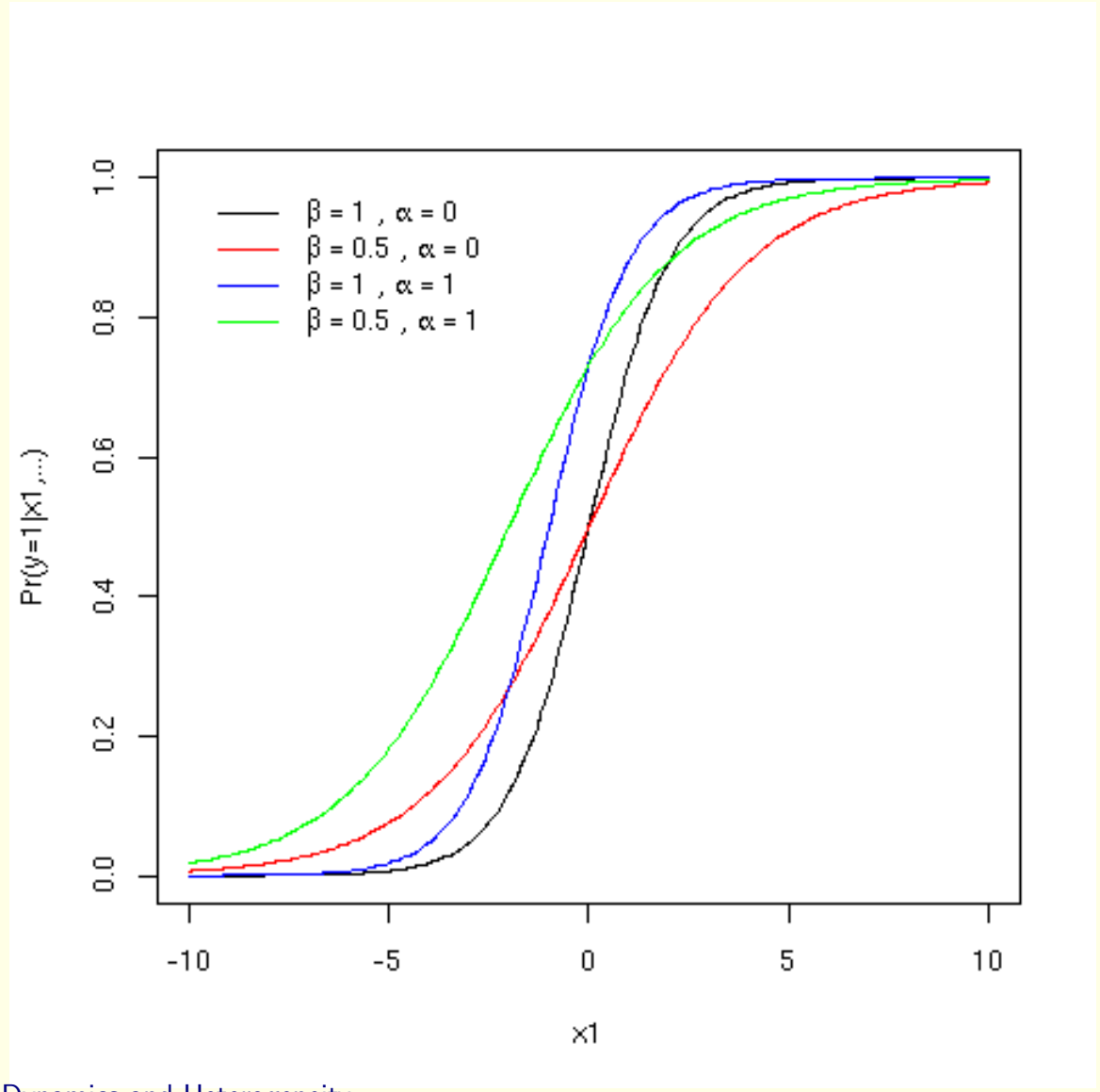
Most of the distributions described above are not single-parameter distributions. However, there is not really enough information in the data to identify two parameters. As a result, we tend to work with standardized distributions. For example, the probit model [based on the normal] sets $\sigma = 1$. The logit model [based on the logistic distribution] sets $\sigma = \frac{\pi}{\sqrt{3}}$. Because of this fact, we do not actually estimate β, we estimate a scaled parameter $\frac{\beta}{\sigma}$.[2] It also means that $\hat{u}_i^*$ is scaled to σ. $\hat{\pi}$ is unaffected.

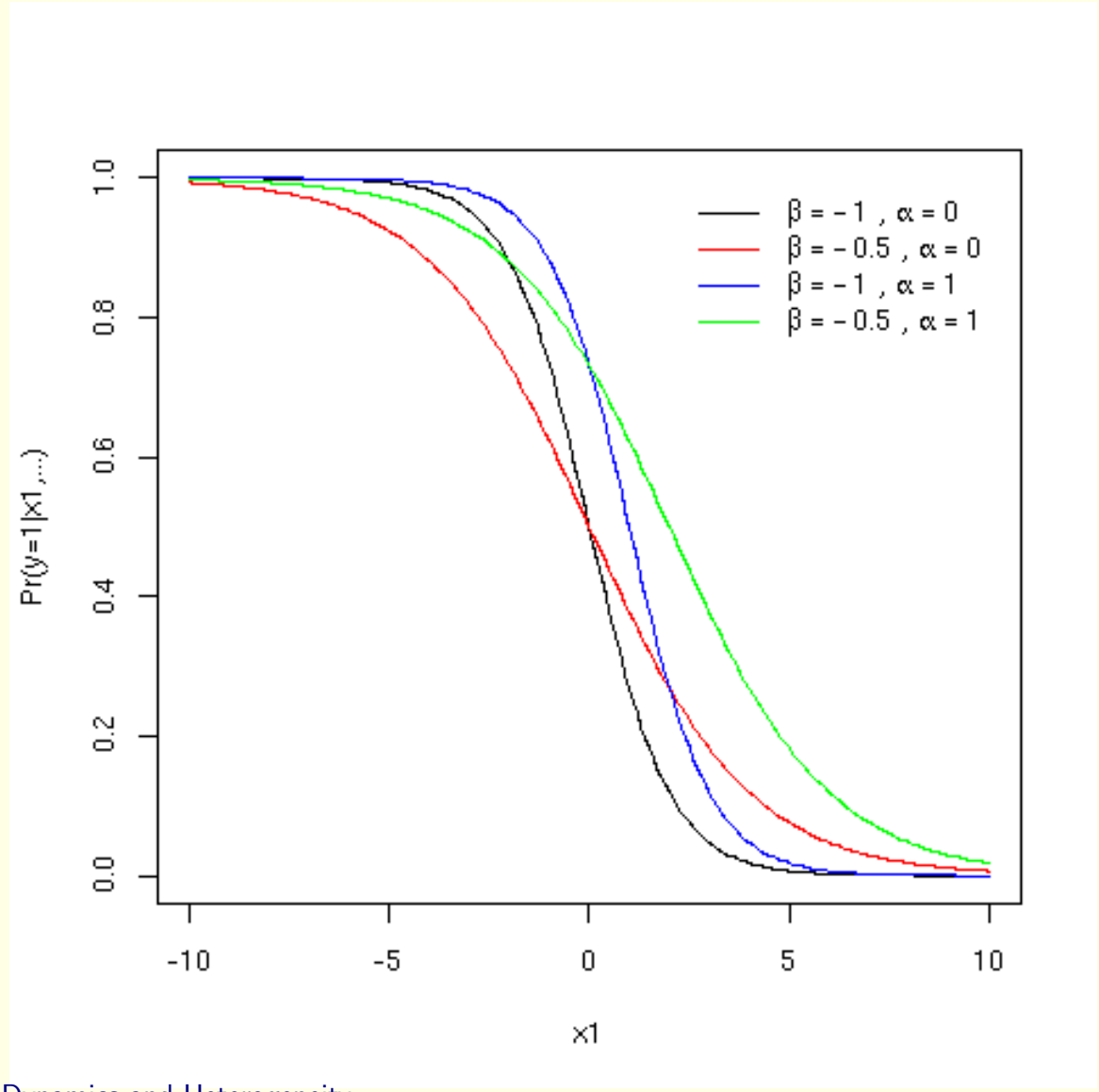
This does not influence hypothesis testing on β nor does it inhibit our ability to uncover consistent estimates of the probabilities – the quantities of interest.

[2] It is for this reason that we can come up with an approximation relating logit and probit estimates of $\hat{\beta}$. The ratio is 1.8, but as Long mentions, others are argued to work better.

Marginal Effects

- Marginal effects should generally be calculated at theoretically interesting [referential] values of the independent variables.
- For nonlinear models, marginal effects depend on decisions about where to set other variables.
- King, et. al. have some software for Stata [clarify] and R [Zelig] to do this for you.
- If you use software to do it, know how they do it. Some of the choices of the programmers are not innocuous.
- Unknowns have sampling distributions; estimated quantities have associated uncertainty and the transparent communication of relevant quantities requires that we pay serious attention to this.


β = 1 , α = 0
β = 0.5 , α = 0
β = 1 , α = 1
β = 0.5 , α = 1
Pr(y=1|x1,...)
0.0
0.2
0.4
0.6
0.8
1.0
-10
-5
0
5
10
x1


Pr(y=1|x1,...)
0.0
0.2
0.4
0.6
0.8
1.0
-10
-5
0
5
10
x1
β = −1 , α = 0
β = −0.5 , α = 0
β = −1 , α = 1
β = −0.5 , α = 1

Some Notation

Reviewing, let us define: $\mathcal{L}(\theta|\vec{y})$ as the likelihood, simplify to $\mathcal{L}(\theta)$.

1. The Score Vector

$$S(\theta) \equiv \frac{\partial \ln \mathcal{L}(\theta)}{\partial \theta}$$

2. The Hessian Matrix

$$H(\theta) \equiv \frac{\partial^2 \ln \mathcal{L}(\theta)}{\partial \theta \partial \theta'}$$

3. The Information Matrix

$$-E\left[\frac{\partial^2 \ln \mathcal{L}(\theta)}{\partial \theta \partial \theta'}\right]^* = I(\theta) \equiv E[S(\theta)S(\theta)']$$

Cramer and Rao's Inequality

Assuming that the density of y satisfies regularity conditions, the variance of an unbiased estimator $\hat{\theta}$ of a parameter θ,

$$V(\hat{\theta}) \geq [I(\theta)]^{-1} = \left(-E \left[\frac{\partial^2 \ln \mathcal{L}(\theta)}{\partial \theta \partial \theta'} \right] \right)^{-1}$$

I will spare you the proof on this. Loosely, the regularity conditions are on the density of the random variable that appears in $\mathcal{L}$ that ensure that the Lindberg-Levy CLT will apply to observations on the random vector $y = \frac{\partial \ln f(x|\theta)}{\partial \theta}$. Among them are finite moments of x to order 3 and that the range of the x's is independent of the parameters.

Properties of MLE's

Consistency: $\operatorname{plim} \hat{\theta}_{MLE} = \theta$

Asymptotic Efficiency:

$$V(\hat{\theta}_{MLE}) = I(\theta)^{-1} = \left(-E \left[\frac{\partial^2 \ln \mathcal{L}(\theta)}{\partial \theta \partial \theta'} \right] \right)^{-1}$$

Asymptotic Normality: $\hat{\theta}_{MLE} \overset{a}{\sim} N(0, I[\theta]^{-1})$
Invariance: MLE of $\gamma = c(\theta)$ is $c(\hat{\theta}_{MLE})$

Asymptotically Equivalent Tests

Define likelihoods $\mathcal{L}(\theta)$ in two variants:
$\mathcal{L}_U(\theta)$ as an **unrestricted** model.
$\mathcal{L}_R(\theta)$ as a restricted model, nested in $\mathcal{L}_U(\theta)$: $\theta_R \subsetneq \theta_U$.

1. Wald: Requires only the computation of $\mathcal{L}_U$
   Can use a robust var/cov matrix.
   Worst small sample properties, see Fears, et. al. (1996) and Pawitan (2000)

2. Lagrange Multiplier: Requires only the computation of $\mathcal{L}_R$
   Can use a robust var/cov matrix.

3. Likelihood Ratio: Requires the computation of $\mathcal{L}_U$ and $\mathcal{L}_R$
   Cannot use a robust var/cov matrix.
   $2(\ln \mathcal{L}_U - \ln \mathcal{L}_R) \sim \chi^2_m$ where m is the number of restrictions. Indeed, doesn't need one and that's good.

A Brief Diversion to GEE

Semi-parametric regression that relies on specifying the first-two moments.

Random effects and variances are nuisance (this is the PA option)

Estimating equations, not likelihoods so no L-R tests.

Specifies the within-group structure.

Uses the sandwich for variance/covariance matrix

Back to the Story

- Is the effect that we want to identify a within or a between effect?
- A brief aside on the population issue
- BIG PICTURE ISSUE: Just because data are not continuous does not mean heterogeneity and dynamics don't matter.

Nuisance v. Substance

There are methods, like GEE, that treat dynamics and other issues as nuisance. Specify some correlation structure and then utilize a moment-based quasi-likelihood estimator to get parameter estimates because we don't care about the correlation. In many instances, this is probably fine. But many times, the precise time issues are essential.

Panel Data Count Models

Stata estimates two panel data count model classes – the Poisson and the Negative Binomial. In most cases, the fact that the Poisson is a single-parameter distribution leads to a default preference for the negative binomial (of which the Poisson is a special case).

- `xtpoisson` estimates fixed effects, random effects, and population averaged versions of the Poisson model for TSCS/CSTS. The population averaged version is simply a GEE with a correlation option.

- `xtnbreg` is similar with a negative binomial regression model. The options are the same.

Binary Models

Stata estimates three panel data binary estimators – the logit, probit, and cloglog. Fixed, random, and population averaged versions exist for logit and cloglog; the probit has no fixed effects version. The PA is, again, a GEE estimator.

Ordered Models

Stata estimates two classes of panel data ordinal regressions – the logit and probit. Both are random effects estimators, though population averaged versions exist. The PA is, again, a GEE estimator.

Mixed Effects Models

`xtmelogit` (logit) and `xtmepoisson` (Poisson) estimate mixed effects regression models for TSCS/CSTS data. The ideas and implementation are similar to the related command `xtmixed` that we have discussed.

`xtgee` **syntax**

`xtgee` operates off of the GLM family and link function ideas. For example, probits and logits are family (binomial) with a probit or logit link. The key issue becomes specifying a working correlation matrix (within-groups/units) from among the options of exchangeable, independent, unstructured, fixed (must be user specified), ar (of order), stationary (of order), and nonstationary (of order).

Day VIII: Dynamic Panel GLM

Binary Dynamics

There are four classes of discrete time series models that we might use for incorporating dynamics for binary observations varying across both time and space. These get some treatment in the paper by Beck, et. al.

- Latent dependence (Dynamic Linear Models)
- State dependence (Markov Processes)
- Autoregressive disturbances
- Duration (survival models and isomorphisms)

Latent Dependence

Carry on the setup from yesterday.

$$u^*_{i,t} = X\beta + \rho u^*_{i,t-1} + \epsilon_{it}$$

This is the analog of a lagged dependent variable regression fit in the latent space rather than the observed data. Such models are probably easiest to fit using Bayesian data augmentation.

Autoregressive Errors and Serial Correlation

$$u^*_{i,t} = X\beta + \epsilon_{it} \tag{13}$$

$$\epsilon_{i,t} = \rho\epsilon_{i,t-1} + \nu_t \tag{14}$$

where ν are i.i.d. The model is odd in the sense that a shock to X dies immediately but a shock to an omitted thing has dynamic impacts. There are some suggested tests for serial correlation. We will implement one of them that employs the generalized residual. The idea is similar to what we have seen before. Here, we have two outcome values and two possible generalized residuals. We either have the density over the CDF or the negative of the density over one minus the CDF. Then we want the covariance in time of the generalized residuals

and need to calculate a variance given as

$$V(s) = \sum_{t=2}^{T} \frac{\phi_t^2 \phi_{t-1}^2}{\Phi_t(1-\Phi_t)\Phi_{t-1}(1-\Phi_{t-1})}$$

We could apply this individually or collectively to the whole set with N summations added to the mix. One can show that the covariance over the square root of $V(s)$ has an asymptotic normal distribution.

BKT 1998

Beck, Katz, and Tucker (1998) point out that BTSCS are grouped duration data. Indeed, a cloglog discrete choice model is a Cox proportional hazards model. They are not similar, like each other, whatever. They are isomorphic. One can leverage this to do something about the temporal evolution of binary processes. Let's get to the details.

Markov Processes

Markov processes extend to a general class of discrete events observed through time and across units. While the reading discusses the binary case, extensions for ordered and multinomial events are straightforward. I will show two examples.

$$\mathbf{P} = \begin{pmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1J} \\ \pi_{21} & \ddots & \dots & \vdots \\ \vdots & \ddots & \dots & \vdots \\ \pi_{J1} & \pi_{J2} & \dots & \pi_{JJ} \end{pmatrix}$$

- Rows represent s^t: the state up to time t
- Columns represent y^t
- Rows sum to unity

The idea is that the current outcome depends on covariates and the prior state. We can do a lot with that.

Some General Comments on Panel GLM

- One has to be careful with these extensions of standard linear models. Ex. Random effects probit and fixed effects logit.
- The orthogonality of the random effects and the regressors is maintained.
- In most cases, the real trouble is incidental parameters. That may not be as harsh as it initially seems. William Greene has an interesting argument about this in his paper, "Estimating Econometric Models with Fixed Effects".

To My Examples

Questions that arise:

- What do the state dependent parameters represent? Interpreting interaction terms.
- Do the effects of a given variable depend on the prior state?
- Is the effect given a prior state differentiable from zero?
- How do we calculate these things?

Types of Missing Data

- OAR (Observed at Random)
  Missingness on Y is not determined by X.

- MAR (Missing at Random)
  Missingness on Y is not determined by Y.

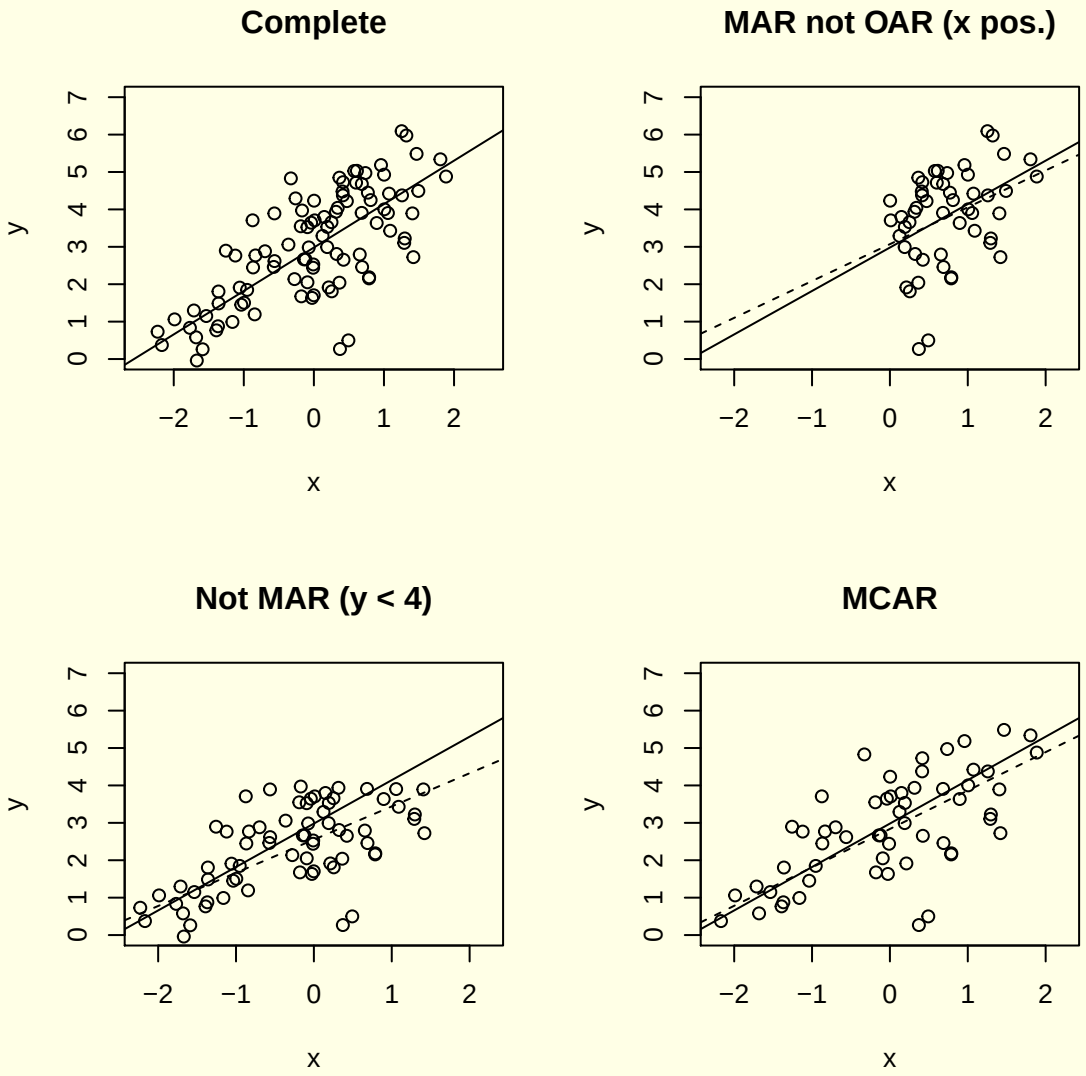
- MCAR (Missing Completely at Random)
  Missingness is both OAR and MAR.

Suppose two variables X and Y.

Illustrating these Points

```
> library(MASS)
> big.X <- mvrnorm(n = 100, c(0, 0, 0), matrix(c(1, 0.8, -0.7, 0.8,
> x <- big.X[, 1]
> y <- 3 + x + rnorm(100)
> df1 <- data.frame(x = x, y = y)
> df2 <- df1
> df2$x[df1$x < 0] <- NA
> df3 <- subset(df1, subset = y < 4)
> df4 <- df1[sample(c(1:100), size=60, replace=FALSE),]
> par(mfrow = c(2, 2))
> plot(df1, main = "Complete", ylim = c(0, 7), xlim = c(-2.5, 2.5))
> abline(lm(y ~ x, data = df1))
> plot(df2, xlim = c(-2.25, 2.25), ylim = c(0, 7), main = "MAR not
> abline(lm(y ~ x, data = df1), lty = 1)
```

```
> abline(lm(y ~ x, data = df2), lty = 2)
> plot(df3, xlim = c(-2.25, 2.25), ylim = c(0, 7), main = "Not MAR
> abline(lm(y ~ x, data = df1), lty = 1)
> abline(lm(y ~ x, data = df3), lty = 2)
> x2 <- big.X[, 2]
> x3 <- big.X[, 3]
> plot(df4, xlim = c(-2.25, 2.25), ylim = c(0, 7), main = "MCAR")
> abline(lm(y ~ x, data = df1), lty = 1)
> abline(lm(y ~ x, data = df4), lty = 2)
```


Complete
MAR not OAR (x pos.)
Not MAR (y < 4)
MCAR
y
x
0 1 2 3 4 5 6 7
−2 −1 0 1 2

What to do?

- Assess missingness?
- Multiple Imputation
- MCMC

Assessing Missingness

- Simple regression techniques (binary GLMs) can serve the useful purpose of identifying the predictors of missingness.
- As a practical matter, we can't really establish the true nature of missingness.
- BUT, we can at least examine missingness as a function of observed covariates. Doing this, we should be careful to engage functional forms above and beyond simple linearity.

Multiple Imputation

- Multiple imputation: the process of "imputing" missing data on multiple occasions to rectangularize a data matrix for analysis.
- Most commonly done with a multivariate normal distribution where the mean vector and variance/covariance matrix form the basis for imputing.
- Other methods are also frequently used including versions of "nearest neighbors" and simple mean imputation. The latter is generally a bad strategy.
- R makes a few methods of imputation readily available.

Imputation with R

- `Amelia`: Performs multiple imputation under a multivariate normal distribution. It contains functions for time series, cross-sectional, and time series cross-sectional data and includes an ability to handle nominal and ordered data.

- `robCompositions`: Multiple imputation for missing data on a simplex or other bounded constraint space.

- `mice`: Multivariate Imputation using Chained Equations. Basically, a Gibbs sampler (a full conditional specification of the imputation equations).

- `mi`: Bayesian specifications of regression models for the imputation of missing data.

Imputation in Stata

Stata was a bit late to the multiple imputation game but has improved considerably in version 12. There is an overarching suite of commands that can be employed with the prefix `mi`. There are some things that Stata cannot (does not) do. For example, there is no method for compositional data; this can be problematic. Of course, the negative correlations can allow unbounded approximations to get close, but that isn't what we really want. It is better to use all of the available information in the data because throwing away information when we are "making it up" guarantees that we make it up less well.

Specifically for TSCS/CSTS data, Honaker and King (2010) in the American Journal of Political Science is on this exact problem. Their software is, however, written in R. You can find the paper from `http://gking.harvard.edu/files/pr.pdf`.

Have Imputations, What do I do?

- The trouble with multiply imputed data come in two forms.
- One is that we want to analyze the (now) complete data but need to account for the presence of predictions.
- The other is that we need some method or methods for the assessment of our imputation.

Analyzing Imputed Data

- Imputed data are predictions from models. Not unlike any predicted variable, the imputations are draws from a sampling distribution and cannot be argued fixed in repeated samples with a straight face. Moreover, the sampling distributions of statistics to which imputations are inputs must reflect the uncertainty of the imputation alongside more conventional uncertainty about the statistic.

- Rubin first gave a formula for combining imputation-based statistics.

- It relies on the asymptotic normality of the statistics; they are a linear combination of normals.

Diagnostics for Imputed Data

- Functions exist for plotting patterns of missingness: `missing.pattern.plot`
- Comparing histograms (after imputation): `mi.hist`, etc.
- Comparing scatterplots: `mi.scatterplot`

The Statistics of it all

We can calculate a few interesting quantities that inform imputation independent of the ultimate goal. For example, let

$$W_\beta = \sum_{k=1}^{K} \frac{V_k}{K}$$

be the average within imputation variance of β_k. Similarly, let the between imputation variance of β be,

$$B_\beta = \sum_{k=1}^{K} \frac{(\beta_k - \overline{\beta})^2}{(K-1)}$$

which yields a total variance of

$$T_\beta = B_\beta(1 + \frac{1}{k}) + W_\beta$$

which gives $\frac{\overline{\beta}}{\sqrt{T_\beta}} \sim t_\nu$ where

$$\nu = (K-1)[1 + WB^{-1}(1 + \frac{1}{K})^{-1}].$$

Furthermore, notice that if the imputations are completely uninformative regarding β, then $\beta_k = \beta^* \; \forall k \in K$ and $T_\beta = W_\beta$. This allows us to construct a ratio, $r = \frac{(1+\frac{1}{K})B_\beta}{W_\beta}$ to measure the increase in variance owing to imputation. Finally, let $\epsilon = \frac{r}{1+r}$ be the proportion of missing information. All of this comes together when the relative efficiency of K imputations relatively to an infinite number is $(1 + \frac{\epsilon}{K})^{-1}$. Take an example of $K = 0.5$ and 5 imputations.

Some General Comments

There is almost no reason not to impute data. Not imputing throws away information. Imputing may not add any but it allows us to retain "real" information. Consider the following scenarios. Suppose that MAR fails. What does imputation really do and what are the properties of the original estimator anyway? What happens with imputation when MAR holds? What is your belief that MCAR ever holds?

The Bayes Approach

Missing data are like anything else that is missing. In the Bayesian framework, we want to sample the missing quantities. We can learn about their distributions, covariances, and the like but we fundamentally want to learn as little as possible from the missing data but maximize the quantity of information recovered from other nonmissing things. Imputation simply becomes another part of the sampler.