Lecture Notes 9

Econ B2000, Statistics and Introduction to Econometrics

Kevin R Foster, the City College of New York, CUNY

Fall 2014

Binary Dependent with Trees and Forests

With Tree Models (from computer science) the emphasis is on prediction not necessarily causation. This can make economists crazy although it can also be a good way to get at causation – are there certain "features" (which is the term that computer science uses instead of 'explanatory variables') that can easily classify some outcome? This can be part of a data description or modeling exercise.

An R program to predict whether a person is covered by employer-provided health insurance is:

```
library('rpart')
# tree model of whether has health insurance
model1 <- rpart(health_ins ~ Age + I(Age^2) + female + AfAm + Asian +
      Amindian + race_oth + Hispanic + educ_hs + educ_smcoll + educ_as +
      educ_bach + educ_adv + married + divwidsep + union_m + veteran +
      immigrant + immig2gen,data = dat_use_hi)
summary(model1)
plot(model1)
text(model1, use.n = TRUE, all=TRUE, cex=.8)
```

We could improve this method by going back to the idea that we discussed with k-nn, where we split into training and evaluation sets – use 80% of the data to train the tree, then see how well it would classify the remaining 20%. This helps if you worry about overfitting.

Random Forests are more complex although they can offer improvements to classification accuracy. They are notoriously difficult to understand or explain, however – they are often mostly a "black box". Nevertheless they can be a useful method of classification even if as a comparison – if a random forest model classifies A% correctly while your preferred model gets B%, then the difference (A-B) can be a useful way to assess how good is the model.

The idea of a Random Forest is to take a randomly-chosen sub-set of the data and build a tree model from it. Then take another randomly-chosen sub-set and build another tree. And another and another... Take these trees and aggregate them (perhaps build 10 trees and figure out if 7 imply one outcome whereas 3 imply the other outcome). (These are random subsets of your 80% training set.)

```
# random Forest
library('randomForest')
set.seed(54321)

# the command system.time() tells how long it takes
system.time(model3 <- randomForest(as.factor(health_ins) ~ .,
      data=dat_cps_rf, importance=TRUE, proximity=TRUE))
print(model3)
```

```
round(importance(model3),2)
varImpPlot(model3)
```

The random Forest gives a "Confusion matrix" comparing the ones that are truly 0/1 versus what is predicted:

	actual 0	actual 1
predicted 0	1558	1961
predicted 1	806	9988

The previous logit model gives results of:

	actual 0	actual 1
predicted 0	5363	3415
predicted 1	15261	61094

The numbers of observations are different because I had clipped the size of the data for the random forest in order to economize on computing time. So it's not apples-to-apples but skewed in favor of logit. But if we look at the fraction in each class, we see that:

	random forest		logit model	
	actual 0	actual 1	actual 0	actual 1
predicted 0	0.109	0.137	0.063	0.040
predicted 1	0.056	0.698	0.179	0.718

So the random forest mis-classified 19.3% of the observations while the logit model mis-classified 21.9% - so even with nearly six times more observations, the logit was a worse fit overall. (You can tweak both methods to do better, maybe a forest of conditional inference trees would be better or you can better specify the logit. These results are illustrative.)

Random forests can also be done for regression problems – the dependent variable need not be 0/1 as above but can be a continuous variable.

These methods are still relatively new in economics; see Hal Varian's piece on *Big Data: New Tricks for Econometrics*.

Experiments and Quasi-Experiments

- ideal: double-blind random sort into treatment and base sets
- differences estimator for "natural experiments" or quasi-experiments
- Problems can be internal:
    - incomplete randomization
    - failure to follow treatment protocol
    - attrition
    - experiment (Hawthorne) effects
- or external
    - non-representative sample
    - non-rep program
    - treatment/eligibility
    - general equilibrium effects

Factor Analysis

Another common procedure, particularly in finance, is a factor analysis. This asks whether a variety of different variables can be well explained by common factors. Sometimes when it's not clear about the direction of causality, or where the modeler does not want to impose an assumption of causality, this can be a way to express how much variation is common. As an example, one price that people often see, which changes very often, is the price of gasoline. If you have data on the prices at different gas stations over a long period of time, you would basically see that while the prices are not identical, they move together over time. This is not surprising since the price of oil fluctuates. There might be interesting variation that at some times certain stations might be more or less responsive to price changes – but overall the story would be that there is a common influence.

Factor Analysis (and the related technique of Principal Components Analysis, PCA) are not model-based and can be useful methods of exploration. An example might be the easiest way to see how it works.

I got daily data from Federal Reserve on Eurodollar interest rates for 1-, 3-, and 6-months, from 1971-2014 (so called since it was originally the rate to borrow dollars from a bank in London, which remains the center of this market).

```
prcomp1 <- prcomp(~ ed1m + ed3m + ed6m, data = data_2)
summary(prcomp1)
```

Which shows that the first principal component explains 99.7% of the variation in these interest rates.

(With a wider span of maturities, we often find that 3 factors explain most interest rate movements: level, slope, and curvature.)

Spike & Slab, Lasso, LOESS

There are many other regression techniques.

I should have mentioned LOESS (local estimation with polynomials, not the kind of soil!) back with nonparametric regression, it is a form of that – where we think there is some smooth function y = f(x) but we want to estimate a very generic function, f( ). Unlike the nonparametric estimation previously it is much less computationally intensive (so runs much faster). The main limitation for our purposes is that X can have at most 4 variables, which must all be continuous.

```
model_loess1 <- loess(WSAL_VAL ~ Age,data3)
y_loess1_pred <- predict(model_loess1, data.frame(Age = seq(25, 55, 1)), se
      = TRUE)
plot(seq(25, 55, 1),y_loess1_pred$fit)
```

Lasso and Spike and Slab are both used for selecting which variables are "important" in predicting. Note as usual that important in prediction might not be the same as causal, however again we can explore the data to see. Both techniques will pare off X-variables that do not contribute much predictive value to the regression. In cases where we have very few observations (i.e. most of macro), these would not be appropriate, however in cases with

dense data then it is reasonable to consider – if your variable of interest is not selected for prediction, then you have to think about why.

Much of the impetus for developing these sorts of models comes from either websites (that get arrays of data streaming through, and try to figure out which have any predictive value) or genomics (which have huge numbers of candidate genetic markers, and try to figure out which have predictive value).

Lasso is Least Absolute Shrinkage and Selection Operator, and in R is usually implemented with the `lars` package.

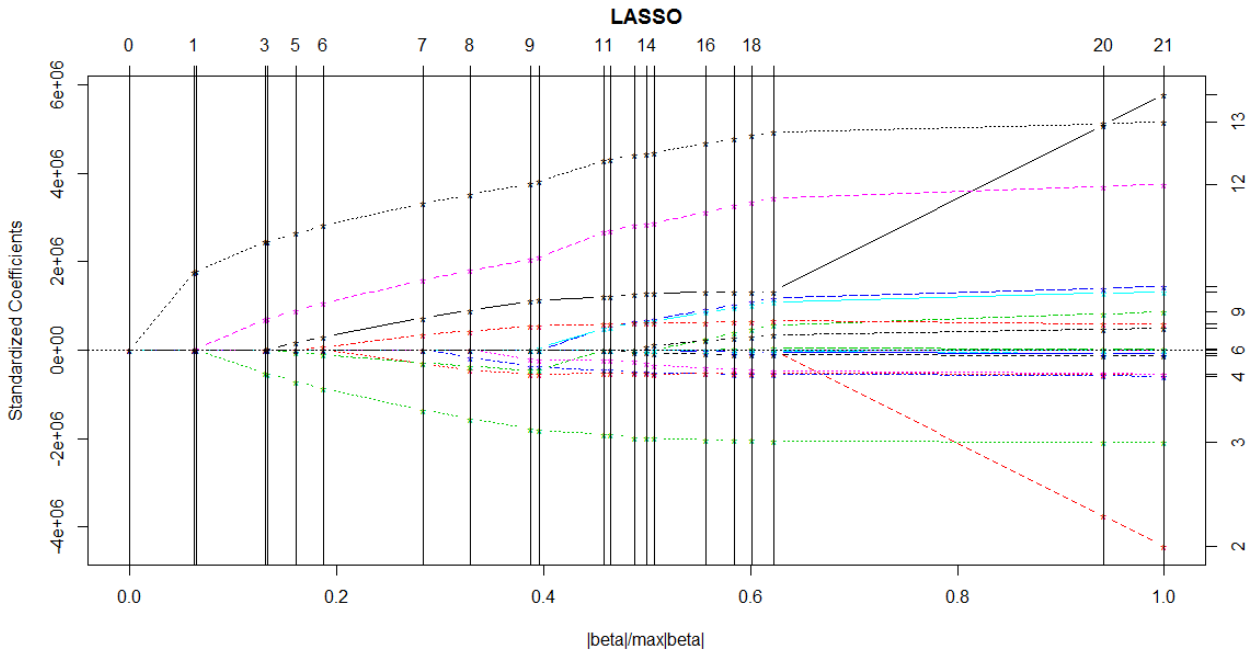
This finds coefficients that not only minimize the squared residuals (just like OLS) but also tries to minimize the squared coefficient sizes – so it penalizes 'too many' explanatory variables. In machine learning this is a way of finding efficient predictors but for our purposes it helps to see which variables are important in the model.

```
x_varb <- cbind(Age,I(Age^2), female, AfAm, Asian, Amindian,race_oth,
      Hispanic, educ_hs, educ_smcoll, educ_as, educ_bach, educ_adv,
      married, divwidsep, union_m, veteran, immigrant, immig2gen)
require(lars)
model_lars <- lars(x_varb,WSAL_VAL)
summary(model_lars)
plot(model_lars)
coef(model_lars)
```

We can get an idea of how it classifies the importance of the different factors from our basic wage regression,

educ_adv	educ_bach	female	Age	educ_hs	married	Hispanic	AfAm	immigrant	educ_as	educ_smcoll	race_oth	immig2gen	union_m	divwidsep	Asian	veteran	Age-sqr	Amindian
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22954	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23257	243	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31859	7169	-4909	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32155	7420	-5081	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34607	9450	-6668	95	-604	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36714	11223	-8001	164	-1133	775	0	0	0	0	0	0	0	0	0	0	0	0	0
43290	16693	-12588	387	-3038	3338	-3830	0	0	0	0	0	0	0	0	0	0	0	0
45911	18857	-14400	481	-3842	4165	-5680	-2586	0	0	0	0	0	0	0	0	0	0	0
49139	21393	-16570	594	-4839	5237	-6796	-5504	-2361	0	0	0	0	0	0	0	0	0	0
49742	21916	-16842	607	-4738	5350	-6872	-5821	-2605	643	0	0	0	0	0	0	0	0	0
55913	27907	-17698	645	0	5654	-6320	-6749	-2580	6967	6179	0	0	0	0	0	0	0	0
56308	28266	-17835	652	0	5708	-6328	-6911	-2667	7383	6575	0	0	0	0	0	0	0	0

57685	29520	-18317	675	0	5881	-6365	-7518	-2964	8843	7971	-1336	0	0	0	0	0	0	0
58011	29829	-18445	682	0	5937	-6524	-7675	-3616	9225	8334	-1749	704	0	0	0	0	0	0
58231	30034	-18530	687	0	5973	-6629	-7778	-4050	9479	8576	-2017	1169	-261	0	0	0	0	0
60960	32742	-18765	698	2688	6056	-6531	-8020	-4736	12304	11356	-2694	2219	-945	0	0	0	0	0
62411	34182	-18899	702	4109	6227	-6479	-8132	-5105	13796	12823	-3048	2783	-1308	290	0	0	0	0
63239	35000	-18971	703	4913	6313	-6517	-8220	-5265	14639	13653	-3205	3153	-1511	442	-261	0	0	0
64286	36035	-19059	705	5929	6422	-6566	-8332	-5466	15704	14702	-3406	3621	-1768	631	-591	44	0	0
66830	38745	-19258	2717	8655	5853	-6751	-8799	1-6257.33	18470	17470	-3955	74943.6444	-2417	292	-1398	141	-25	0
67303	39249	-19296	3089	9160	5749	-6786	-8883	3-6403.30	18983	17984	-4055	75188.6771	-2536	230	-1546	159	-29	144



Spike and Slab (the name refers to the Bayesian prior distributions about coefficients) is implemented in R with the `spikeslab` package. Scott and Varian (2012) refer to the "fat regression" problem where there are more possible explanatory variables than there are observations – there is a severe problem with degrees of freedom. The "spike" refers to the probability that a particular variable is in the model (there is either a 0 or a 1 to select that particular explanatory variable) while the "slab" is the information from the coefficient estimates.

This is another way to gauge the importance of various parts of your model, particularly in cases if there are lots of interactions.

A linear regression with a lot of interactions (returning to our usual CPS wage regression) could include this,

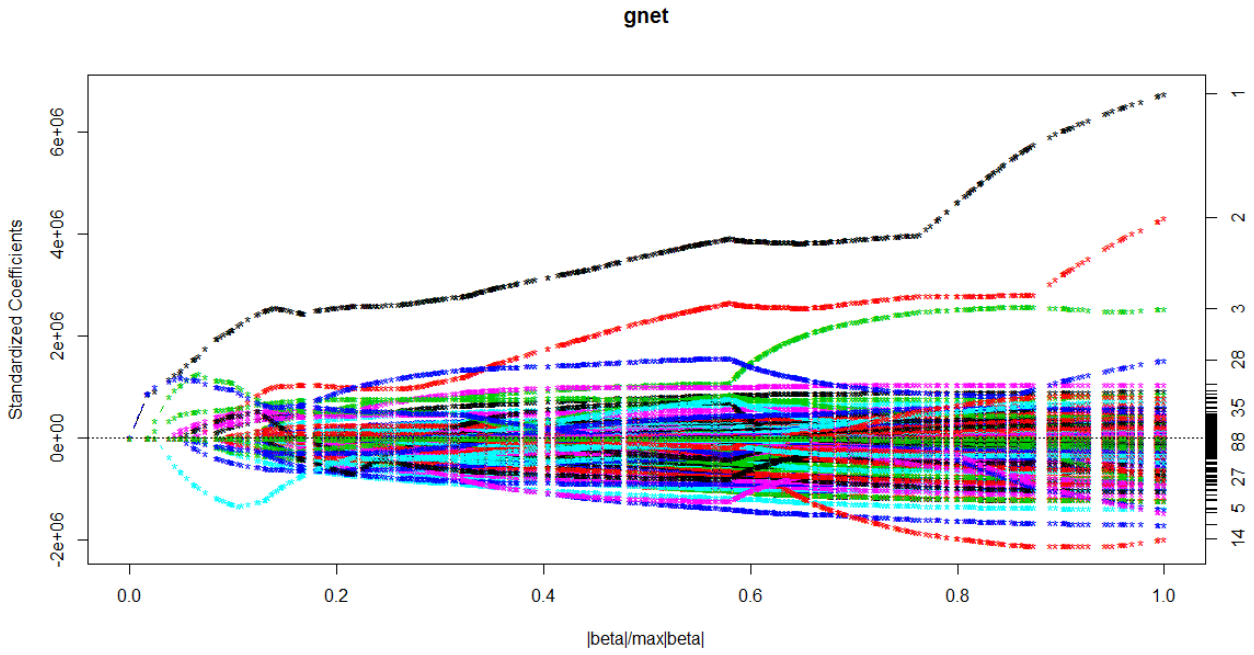
```
modelcompare <- lm(WSAL_VAL ~ (Age + I(Age^2) + female + AfAm + Asian +
      Amindian + race_oth + Hispanic + educ_hs + educ_smcoll + educ_as +
      educ_bach + educ_adv + married + divwidsep + union_m + veteran +
      immigrant + immig2gen) ^2 + (industry_f + occupatn_f +
      state_f)*female, data = dat_8)
summary(modelcompare)
```

Whereas a version with spike and slab would use this code,

```
require(spikeslab)
set.seed(54321)
model1_spikeslab <- spikeslab(WSAL_VAL ~ (Age + I(Age^2) + female + AfAm +
      Asian + Amindian + race_oth + Hispanic + educ_hs + educ_smcoll +
      educ_as + educ_bach + educ_adv + married + divwidsep + union_m +
      veteran + immigrant + immig2gen) ^2 + (industry_f + occupatn_f +
      state_f)*female,  data = dat_8)
summary(model1_spikeslab)
print(model1_spikeslab)
plot(model1_spikeslab)
```

Both will keep your computer running for a while! Note the "set.seed" sets the random number generator so that, if you try it again, you'll get the same output as I did.

The picture is tough to interpret given so many lines,



Other than that there are only a few that really stand out. The "print" call will give the coefficient estimates from this model; the top of that print is:

---> Top variables:

	bma	gnet	bma.scale	gnet.scale
Age:educ_adv	16690.17	17715.13	1123.054	1192.021
Age:educ_bach	8792.73	11872.62	485.761	655.912
Age	7020.34	7152.103	817.248	832.587

occupatn_f17	-6143.29	-6595.35	-18991.2	-20388.7
occupatn_f8	-5395.99	-5487.32	-23699.6	-24100.8
occupatn_f10	4805.165	4645.705	20435.86	19757.69
occupatn_f6	-4621.3	-4926.29	-33460.7	-35668.9
occupatn_f21	-4546.55	-4899.63	-18791.1	-20250.4
occupatn_f22	-4533.39	-4921.73	-19702.5	-21390.3
female:occupatn_f10	-4424.35	-4457.37	-22250	-22416.1

Where the "bma" (Bayesian Model Averaging) and "gnet" (the generalized elastic net, with penalty parameters for coefficients) refer to different estimation methods; the first two columns are coefficients for the normalized values of the x-variables (with mean 0 and std dev 1) while the last two columns are the usual coefficient estimates.

From looking at the top ones most likely to be selected for inclusion in the model, we see that the first 2 most important variables are age interacted with education measures, then age, then various occupation categories. This is similar to the LASSO that implied that education was most important.

*(If you learn nothing else from this course, learn that the data show that education is important! Although, you know, probably because people with more education actually learn and remember the s*** that their professors say…)*

Time Series

Basic definitions:

- first difference $\triangle Yt = Yt - Yt\text{-}1$
- percent change is $\%\Delta Y_t = \frac{\Delta Y_t}{Y_{t-1}}$ and is approximately equal to ln(Yt) – ln(Yt-1) – this log approximation is commonly used
- lags: the first lag of Y_t is Y_{t-1}; second lag is Y_{t-2}, etc.
- Autocorrelation: how strong is last period data related to this period? The autocorrelation coefficient is $\rho_j = \frac{\text{cov}\left(Y_t, Y_{t-j}\right)}{\text{var}\left(Y_t\right)}$ for each lag length, j. Sometimes plot a graph of the autocorrelation coefficients for various j.
- Common assumption: Stationarity: a model that explains Y doesn't change over time – the future is like the past, so there's some point to examining the past – a crucial assumption in forecasting! But this is why we usually use stock returns not stock price – the price is not likely stationary even if returns are. (Also often assume ergodic.)
- If autocorrelations are not zero, then OLS is not appropriate estimator if X and Y are both time series! The standard errors are a function of the autocorrelation terms so cannot properly evaluate the regression.
- Seasonality is basically a regression with seasons (months, days, whatever) as dummy variables. So could have $Y_t = \beta_0 + \beta_1 January + \beta_2 February + \beta_3 March + \ldots + \beta_{11} November + u_t$ - remember to

leave one dummy variable out! Or
$Y_t = \beta_0 + \beta_1 Monday + \beta_2 Tuesday + \ldots + \beta_{11} Saturday + u_t$.

Types of Models

- AR(1) – autoregression with lag 1
- $Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$
- Forecast error is one-step-ahead error
- Note that can re-write the AR(1) equation, by substituting $Y_{t-1} = \beta_0 + \beta_1 Y_{t-2} + u_{t-1}$, as $Y_t = \beta_0 + \beta_1 \left(\beta_0 + \beta_1 Y_{t-2} + u_{t-1}\right) + u_t = \beta_0 \left(1 + \beta_1\right) + \beta_1^2 Y_{t-2} + \beta_1 u_{t-1} + u_t$, then substitute in for $Y_{t-2} = \beta_0 + \beta_1 Y_{t-3} + u_{t-2}$, and so on. So the current value is a function of all past error terms, $Y_t = \beta_0 \left(1 + \beta_1 + \beta_1^2 + \ldots + \beta_1^T\right) + \left[u_t + \beta_1 u_{t-1} + \beta_1^2 u_{t-2} + \ldots + \beta_1^T u_{t-T}\right] + \beta_1^T Y_{t-1-T}$. Note that as long as $|\beta_1| < 1$, the last term drops and the sums converge as $T \to \infty$.
- Reminder of convergent series: look at $\left(1 + \beta_1 + \beta_1^2 + \ldots + \beta_1^T\right)$, note that $\beta_1 \left(1 + \beta_1 + \beta_1^2 + \ldots + \beta_1^T\right) = \left(\beta_1 + \beta_1^2 + \ldots + \beta_1^{T+1}\right)$. Add and subtract β_1^{T+1} and fiddle the parentheses to write $\left(1 + \beta_1 + \beta_1^2 + \ldots + \beta_1^T\right) = 1 + \left(\beta_1 + \beta_1^2 + \ldots + \beta_1^T + \beta_1^{T+1}\right) - \beta_1^{T+1}$. Notate that ugly term $\left(1 + \beta_1 + \beta_1^2 + \ldots + \beta_1^T\right) = Z$, then the equation says that $Z = 1 + \beta_1 Z - \beta_1^{T+1}$. Solve, $Z - \beta_1 Z = Z\left(1 - \beta_1\right) = 1 - \beta_1^{T+1}$, and $Z = \frac{1 - \beta_1^{T+1}}{1 - \beta_1}$. Substitute this into the previous equation for Y_t
- $Y_t = \beta_0 \frac{1 - \beta_1^{T+1}}{1 - \beta_1} + \left[u_t + \beta_1 u_{t-1} + \beta_1^2 u_{t-2} + \ldots + \beta_1^T u_{t-T}\right] + \beta_1^T Y_{t-T}$. As $T \to \infty$, the first term goes to $\beta_0 \frac{1}{1 - \beta_1}$, the last term goes to zero, and the middle term is $\sum_{\tau=0}^{\infty} \beta_1^\tau u_{t-\tau}$.
- If $\beta_1 = 1$ then none of the terms converge – the model becomes a random walk or integrated with order 1, I(1) or has a unit root. (Can test for this, most common is Augmented Dickey-Fuller ADF.)
    - Also random walk with trend, so $Y_t = \beta_0 + \gamma t + Y_{t-1} + \varepsilon$
    - And random walk with drift, so $Y_t = \beta_0 + Y_{t-1} + \varepsilon$ (but no trend)
    - Or just plain random walk, $Y_t = Y_{t-1} + \varepsilon$
- Random walk means that AR coefficients are biased toward zero, the t-statistics (and therefore p-values) are unreliable, and we can have a "spurious regression" – two time series that seem related only because both increase over time. Consider this case of variables X and Y, each of which are $Z_t = 1 + Z_{t-1} + \varepsilon$ where ε is a random draw from a normal distribution.

```
rm(list = ls(all = TRUE))

const_term <- 1
ar_coeff <- 1
start_val <- 100
```

```
num_terms <- 100

x_val <- matrix(data = NA, nrow = num_terms, ncol = 1)
y_val <- matrix(data = NA, nrow = num_terms, ncol = 1)

x_val[1] <- start_val
y_val[1] <- start_val

set.seed(12345)
x_rand <- rnorm(num_terms, mean = 0, sd = 1)
y_rand <- rnorm(num_terms, mean = 0, sd = 1)

for (indx in 2:num_terms) {
  x_val[indx] <- ar_coeff*x_val[indx - 1] + const_term + x_rand[indx]
  y_val[indx] <- ar_coeff*y_val[indx - 1] + const_term + y_rand[indx]
}

model1 <- lm(y_val ~ x_val)
summary(model1)

(ar(y_val)) #AR method
```

- AR(p) – autoregression with lag p
- $Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + u_t$
- ADL(p,q) – autoregressive distributed lag model with p lags of dependent variable and q lags of an additional predictor, X.
- Need usual assumptions for this model
- Lag length? Some art; some science! Various criteria (AIC, BIC, given in text) to select lag length.
- Granger Causality – jargon meaning that X helps predict Y; more precisely X does not Granger-cause Y if X does not help predict Y. If X does not help predict Y then it cannot cause Y.
- Trends provide non-stationary models
- Random walk non-stationary model:
- Breaks can also give non-stationary models
- test for breaks, sup-Wald test
- Cointegration "The Definitive Overview", ftp://ftp.econ.au.dk/creates/rp/14/rp14_38.pdf
- Can model time series as regression of Y on X, of ln(Y) on ln(X), of ΔY on ΔX, or of %ΔY on %ΔX (where, recall, %ΔY = ΔlnY since the derivative of the log is the reciprocal) – this is where the art comes in!
- Distributed lag models can be complicated (Chapter 15) and so we want at a minimum Heteroskedasticy and Autocorrelation Consistent (HAC) errors – like the heteroskedasticity-consistent errors before (Newey-West)
- VAR – Vector AutoRegression, incorporate k regressors and p lags so estimate as many as k*p coefficients – these are classic in macro modeling, following work of Chris Sims

- GARCH models – Generalized AutoRegressive Conditional Heteroskedasticity models – allow the variance of the error to change over time, depending on past errors – allows "storms" of volatility followed by quiet (low-variance)
    - $y_t = \sigma_t \varepsilon_t;\ \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$ GARCH(p,q)
    - Combine with random walk analysis for IGARCH, etc

In R: read "Time Series Analysis with R" for a high-level overview of what's possible – that has refs to various packages that you can study, as you figure out what exactly you want to do. http://www.stats.uwo.ca/faculty/aim/tsar/

Methodology

As you get more experience with econometrics you can start to understand the old jokes about why the discipline name includes "con" and "trics"! Ed Leamer has a classic paper, *Let's Take the 'Con' Out of Econometrics*. Diedre McCloskey has been a persistent critic, e.g. in *Knowledge and Persuasion in Economics* or *The Trouble with Mathematics and Statistics in Economics*. Chris Sims wrote, *Why are Econometricians so Little Help?* Although Angrist and Pischke wrote *Mostly Harmless Econometrics*. You can understand why so many econometricians advise, "beware of econometricians."

More…

Econometrics goes on and on – there are thousands of techniques for new situations and new conditions, especially now that computing power quickly increases the amount of calculations that can be done. There is so much to learn!