Problems with Predictors

Quantitative Methods II for Political Science
Kenneth Benoit

February 4, 2009
To determine the probability of an event that a random variable $X$ with a normal distribution is less than or equal to $x$, we evaluate the *cumulative distribution function* of the normal probability distribution at $x$.

The *cdf* of the normal distribution is:

$$
\Phi_{\mu, \sigma^2}(x) = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \, dx
$$

We use this integration, which is done numerically by R, to find the total area up to a critical value or between two critical values.

Only difference with the $t$ distribution is you also need degrees of freedom for $t$. 

Each distribution in R, e.g. *norm*, has four forms d, p, q, r

- **density** this computes the curve, for instance:

```r
> x <- seq(-4, 4, 0.1)  # numeric sequence every .1 from 4 to 4
> plot(x, dnorm(x), type="l")  # plot (standard) normal function for x
```

- **distribution** or probability function returns the area under the curve for a given quantile

```r
> pnorm(1.96)  # area up to 1.96 (one-tailed)
[1] 0.9750021
> 2*pnorm(-abs(1.959964))  # area below and above 1.96 (two-tailed)
[1] 0.05
```
Each distribution in R, e.g. \textit{norm}, has four forms \(d, p, q, r\)

- \textbf{quantile} returns the score associated with a given \(p\) value
  
  ```r
  > qt((1-.05/2), df=462)  # quantile functions
  [1] 1.965112
  > qt((1-.05/2), df=30)
  [1] 2.042272
  ```

- \textbf{random} generates draws from the distribution
  
  ```r
  > rbinom(20,2,.5) # count heads from 20 tosses of 2 coins w/Pr(heads)=.5
  [1] 2 0 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1
  ```
Q-Q plots in more detail

▶ Quantile-quantile normal (qqnorm()) plots are so named because they plot the ranked sample quantiles from our distribution against a similar number of ranked quantiles taken from a normal distribution

▶ If the sample is normally distributed, then the line will be straight

▶ We can compute the sequence of probability points using ppoints() which produces a symmetric sequence in $[0,1]$:

$$
\frac{1}{N + 1} : N - 0.5
$$

and which is used to generate the set of probabilities at which to evaluate the inverse normal distribution (for “theoretical quantiles”)
Reconstructing the Q-Q plot in R

```r
> rae <- read.csv("rae2008politics.csv")
> staff.sorted <- sort(rae$staff) # sort the sample data
> y <- quantile(staff.sorted, ppoints(staff.sorted))
> x <- qnorm(ppoints(staff.sorted))
> par(mfrow=c(1,2))
> plot(y~x, main="\"By hand\" Q-Q Norm")
> qqnorm(staff.sorted, main="Built-in Q-Q Norm")
```
General principles for specification

- *Theory* is our best guide
- If the residuals from a model are not significantly different from what might have occurred by chance, then conclude that the model is “mis-specified” (that nothing is going on)
- Tests for misspecification are OK when used judiciously
- We can set aside a subset of observations to be used for testing by making out-of-sample predictions
- Some authors advocate reporting the results of other specifications (a form of “sensitivity analysis”) although this is done rarely, if at all in political science
Common tests for misspecification

- **Tests for omitted variables.** This include $F$ tests and $t$ tests for whether coefficients are individually or jointly zero.
- **RESET: Regression specification error tests.** Tests whether unknown variables have been omitted from a regression specification.
- **Tests for functional form.** These include tests for recursive residuals, the rainbow test, and others (below).
- **Tests for structural change.** To test whether parameters change, such as the Chow test, cumsum, and cumsum-of-squares tests.
- **Tests for outliers.** Cook outlier tests for instance, although there are many others.
- **Tests for non-spherical errors.** Example: Durbin-Watson test. Next week!
- **Tests for exogeneity.** Hausman tests.
- **Others (see Kennedy)**
Error in covariates

- This is the case where we have random error in covariates $X$
- Example: assume we measure spending with (normally distributed) random noise with mean=$0$, sd=$5,000$:

```r
> ## Illustrate problem of random error in X
> require(foreign)
> d <- read.dta("dail2002.dta")
> # model without error
> m1 <- lm(votes1st ~ spend_total, data=d)
> # model with error
> d$spenderror <- d$spend_total + rnorm(length(d$spend_total), 0, 5000)
> m1e <- lm(votes1st ~ spenderror, data=d)
> coef(m1)
  (Intercept) spend_total
683.7550298  0.2336056
> coef(m1e)
  (Intercept) spenderror
1434.9778983  0.1858967
```
# plot the difference in slopes
plot(votes1st ~ spend_total, data=d, xlab="Spending", ylab="Votes")
abline(m1)
points(votes1st ~ spenderror, data=d, pch="+", col="red")
abline(m1e, col="red")
A fix for error in covariates: SIMEX

- SIMEX refers to “simulation-extrapolation” (see Cook and Stefanski)
- Basic idea is straightforward:
  - if a coefficient is biased by measurement error, then adding more measurement error should increase the degree of this bias
  - by adding successive levels of measurement error through simulation, we can estimate the trend of bias due to measurement error versus the variance of the added measurement error
  - using the trend, we then extrapolate back to the case where measurement error is absent
- Note: this assumes you know the variance inflation factor, although you seldom do
- But corrections can be used to test “worst-case” scenarios
SIMEX technical steps

1. Generate additional random pseudo errors from normal distribution $N(0, \zeta_m \sigma_u^2)$ and add to original data

2. Conventionally, we use $\zeta \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$; constraint is that $0 = \zeta_1 < \zeta_2 < \cdots < \zeta_M$

3. Then simulate $m$ data sets with increasingly larger measurement error variances. The total measurement error variance in the $m^{th}$ data set is $\sigma_u^2 + \zeta_m \sigma_u^2 = (1 + \zeta_m)\sigma_u^2$.

4. Repeat a large number of times, and take average for each level of contamination

5. Fit an extrapolant function to “zero-error” case corresponding to $(1 + \zeta_m)\sigma_u^2 = 0$, i.e. $\zeta_m = -1$.
(The extrapolant is usually quadratic.)
> # show difference in SIMEX
> library(simex) # must first install from CRAN
> m1e2 <- lm(votes1st ~ spenderror, data=d, x=TRUE)
> m1.simex <- simex(model=m1e2, SIMEXvariable="spenderror",
+                      measurement.error=5000)
> coef(m1.simex)
  (Intercept)    spenderror
   887.0268191   0.2260427
> coef(m1)
  (Intercept)    spend_total
   683.7550298   0.2336056
> coef(m1e)
  (Intercept)    spenderror
 1434.9778983   0.1858967
SIMEX plot

plot(m1.simex)
Changes of scale

- In short: Linear rescaling of variables will not change the essential key statistics for inference, just their scale.

- Suppose we reexpress $x_i$ as $(x_i + a)/b$. Then:
  - $t$, $F$, $\hat{\sigma}^2$, $R^2$ unchanged
  - $\hat{\beta}_i \rightarrow b\hat{\beta}_i$

- Suppose we rescale $y_i$ as $(y_i + a)/b$. Then:
  - $t$, $F$, $R^2$ unchanged
  - $\hat{\sigma}^2$ and $\hat{\beta}_i$ will be rescaled by $b$

- Standardized variables and standardized coefficients: where we replace the variables (all $x$ and $y$) by their standardized values $(x_i - \bar{X})/SD_x$ (e.g. for $x$). Standardized coefficients are sometimes called “betas”.
More on standardized coefficients

Consider a standardized coefficient $b^*$ on a single variable $x$.

- Formula: $b^* = b \frac{SD_x}{SD_y}$
- Interpretation: the increase in standard deviations of $y$ associated with a one standard deviation increase in $x$
- Motivation: “standardizes” units so we can compare the magnitude of different variables’ effects
- In practice: serious people never use these and you should not either
  - too tricky to interpret
  - misleading since suggests we can compare apples and oranges
  - too dependent on sample variation (just another version of $R^2$)
- We can illustrate this in R, if we use the `scale()`[,1] command to standardize the variables, which transforms them into $z_i = (x_i - \bar{X})/SD_x$
Standardized coefficients illustrated

```r
> dc <- d[complete.cases(d$votes1st, d$spend_total),] # remove missing
> m1.std <- lm(scale(dc$votes1st)[,1] ~ scale(dc$spend_total)[,1])
> coef(m1)
(Intercept) spend_total
683.7550298 0.2336056
> coef(m1.std)
(Intercept) scale(dc$spend_total)[, 1]
-1.100854e-16 7.395996e-01
> coef(m1)[2]*sd(dc$spend_total)/sd(dc$votes1st)
spend_total
0.7395996
```
Collinearity

- When some variables are exact linear combinations of others then we have exact collinearity, and there is no unique least squares estimate of $\beta$.

- When $X$ variables are correlated, then we have (multi)collinearity.

- Detecting (multi)collinearity:
  - look at correlation matrix of predictors for *pairwise* correlations
  - regress $x_k$ on all other predictors to produce $R^2_k$, and look for high values (close to 1.0)
  - Examine eigenvalues of $X'X$
Define:

\[ S_{x_jx_j} = \sum_i (x_{ij} - \bar{x}_j)^2 \]

then

\[ \text{Var}(\hat{\beta}_j) = \sigma^2 \left( \frac{1}{1 - R^2_j} \right) \frac{1}{S_{x_jx_j}} \]

So collinearity’s main consequence is to reduce the efficiency of our estimates of \( \beta \)

So if \( x_j \) does not vary much, then \( \text{Var}(\hat{\beta}_j) \) will be large – and we can maximize \( S_{x_jx_j} \) by spreading \( X \) as much as possible

We call this factor \( \frac{1}{1 - R^2_j} \) a variance inflation factor (the faraway package for R has a function called \texttt{vif()} you can use to compute it)

Orthogonality means that variance is minimized when \( R_j^2 = 0 \)
Additional topics

- Simultaneous equations
  - Identification
  - Indirect least squares
  - Instrumental variables
  - Two-stage least squares
  - Limited-information Maximum Likelihood

- Homework 4

- Replication projects