Regression III
Lecture 4: Linearity Diagnostics

Dave Armstrong

University of Western Ontario
Department of Political Science
Department of Statistics and Actuarial Science (by courtesy)

e: dave.armstrong@uwo.ca
w: www.quantoid.net/teachicpsr/regression3/

Outline for Linearity Discussion

1. The linearity assumption
2. Diagnosis of un-modeled non-linearity (CR Plots, Smoothers)
3. Simple remedies for un-modeled non-linearity (transformations, polynomials).
4. More complicated remedies for un-modeled non-linearities (splines, ALSOS).
   - For their own sake in modeling non-linearities.
   - For use in testing theories about functional form.

The Linearity Assumption

Perhaps the most important assumption of the linear model is that the relationship between y and x is accurately described by a line.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

This allows us to:

1. Characterize the relationship between y and x with a single (or small set of) numbers.
2. Easily interpret the marginal effect of x.
3. Easily present the results of the modeling enterprise.

Diagnosing Non-Linearity

We are often interested in the extent to which data we observe follow the assumption of linearity.

- Binary variables are always linearly related to the observed variables (two points define a line)
- Binary regressors operationalizing a single categorical variable allow for any type of non-linearity to be modeled, leaving no un-modeled non-linearity.
- Continuous (and quasi-continuous) variables are not always linearly related to the response and present opportunities for un-modeled non-linearity.
    - We want to know the extent to which these variables exhibit linear relationships.

Linearity and Multi-Category Variables

Multi-category variables are generally not problematic because we code them as a series of dummy regressors. Thus, we are not imposing any functional form on the relationship between the categories and the response variable.

The waters are a bit murkier for ordinal variables (e.g., state repression or political ideology).

- These variables are often operationalized with relatively few categories.
- However, we often have a strong suspicion that the relationship between these variables and the response is "roughly linear".
    - If the relationship is *not* linear and we represent it with a line, then we are getting a *biased* estimate of the relationship.
    - If the relationship could be represented linearly, and we represent it with a series of dummy regressors, we are getting estimates that are *inefficient*

Testing the Hypothesis

Consider the model[1]:

$$y = f(x) + \varepsilon$$

Ultimately, we want to test whether a linear approximation is sufficient.

$$\begin{aligned} H_0 &: f(x) = \beta_0 + \beta_1 x \\ H_A &: f(x) \neq \beta_0 + \beta_1 x \quad \text{(i.e., the function is more complicated)} \end{aligned}$$

We don't have to have know or specify the functional form of the alternative hypothesis, rather just that it is more complicated than linear.

[1] Covariates can be added to the model below without loss of generality

Testing the Hypothesis: Ordinal Variables

The hypothesis suggested above is relatively easy to test when the independent variable is ordinal (i.e., categorical).

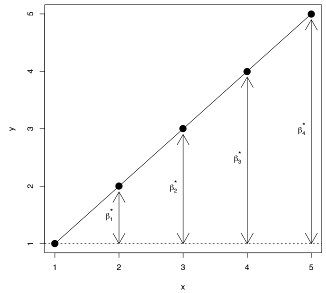
$$\begin{aligned} H_0 &: f(x) = \beta_0 + \beta_x \\ H_A &: f(x) = \beta_0 + \beta_1^* I(x=2) + \beta_2^* I(x=3) + \beta_3^* I(x=4) + \beta_4^* I(x=5) \end{aligned}$$

where $I()$ is an indicator function such that:

$$I(x=k) = \begin{cases} 1 & \text{if } x = k \\ 0 & \text{otherwise} \end{cases}$$

Expectations

Consider the model: $y = \alpha + \beta x + \varepsilon$ where $x = \{1, 2, 3, 4, 5\}$. What would we expect if x and y are perfectly linearly related?



$$\beta_2^* = 2\beta_1^*$$
$$\beta_3^* = 3\beta_1^*$$
$$\beta_4^* = 4\beta_1^*$$

An Example

I generated data with the following such that $x_i \in \{1, 2, 3, 4, 5\}$ and

$$y_i = 2 + x + \varepsilon_i$$

where $\varepsilon_i \sim N(0, 2)$.

We can use an F-test to get the desired result. To accomplish this, we need to do:

1. Run the model by creating dummy variables for all but the smallest category of the variable in question.
2. Test the appropriate restrictions on the model.

Example Continued

Here is the model output:

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5335 -1.2756 -0.0546  1.3060  6.6972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1808     0.1945  16.354  < 2e-16 ***
## x2            0.6041     0.2751   2.196   0.0285 *
## x3            2.0601     0.2751   7.490 3.19e-13 ***
## x4            2.7467     0.2751   9.986  < 2e-16 ***
## x5            4.0309     0.2751  14.655  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.945 on 495 degrees of freedom
## Multiple R-squared:  0.3609,Adjusted R-squared:  0.3557
## F-statistic: 69.87 on 4 and 495 DF,  p-value: < 2.2e-16
```

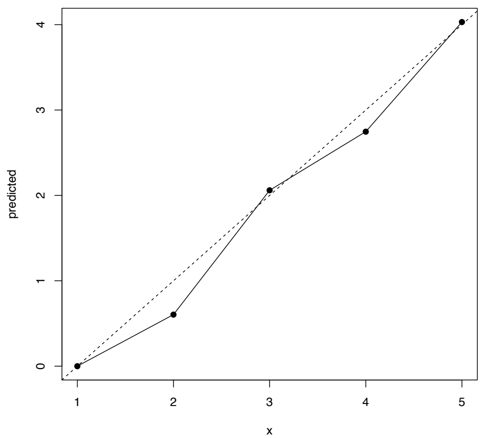

Hypothesis Test

We can also perform a hypothesis test using the general linear hypothesis testing:

```
library(car)
hyps <- c("2*x2 = x3", "3*x2 = x4",
          "4*x2 = x5")
linearHypothesis(mod, hyps)
```

```
## Linear hypothesis test
##
## Hypothesis:
## 2 x2 - x3 = 0
## 3 x2 - x4 = 0
## 4 x2 - x5 = 0
##
## Model 1: restricted model
## Model 2: y ~ x
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    498 1888.4
## 2    495 1872.5  3    15.896 1.4008 0.2418
```

Linear vs. Non-linear effect



Results

The results of the F-test suggest that the dummy variable model is not significantly better than the model with one linear term (i.e., $p > 0.05$).

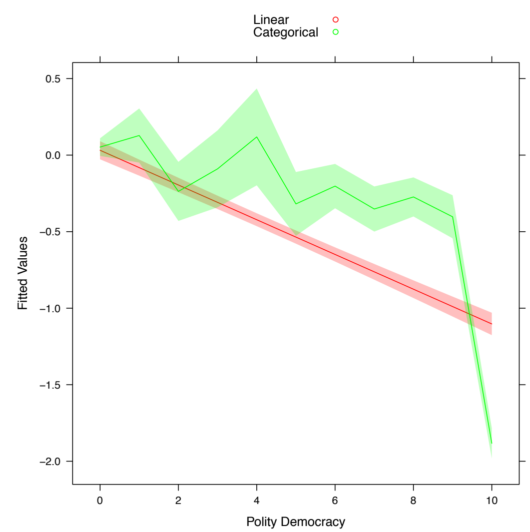
There is another, equivalent way to do this test:

```
restricted.mod <- lm(y ~ as.numeric(x))
unrestricted.mod <- lm(y ~ x)
anova(restricted.mod, unrestricted.mod, test="F")

## Analysis of Variance Table
##
## Model 1: y ~ as.numeric(x)
## Model 2: y ~ x
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    498 1888.4
## 2    495 1872.5  3    15.896 1.4008 0.2418
```


Real Data Example

```
library(foreign)
dat <- read.dta("http://www.quantoid.net/files/reg3/linear_ex.dta")
restricted.mod <- lm(rep1 ~ polity_dem + iwar +
    cwar + logpop + gdppc,data=dat)
dat$polity_dem_fac <- as.factor(dat$polity_dem)
unrestricted.mod <- lm(rep1 ~ polity_dem_fac + iwar +
    cwar + logpop + gdppc,data=dat)
anova(restricted.mod, unrestricted.mod, test="F")

## Analysis of Variance Table
##
## Model 1: rep1 ~ polity_dem + iwar + cwar + logpop + gdppc
## Model 2: rep1 ~ polity_dem_fac + iwar + cwar + logpop + gdppc
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2677 2538.3
## 2   2668 2163.3  9    374.98 51.385 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

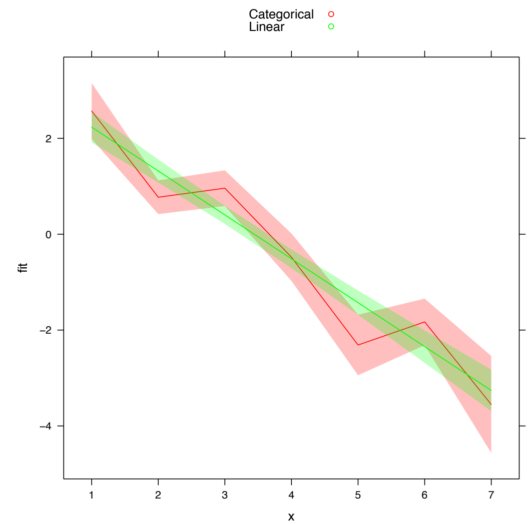

Plot of effects



Linearity of Factors in GLMs

```
library(foreign)
anes <- read.dta("http://www.quantoid.net/files/reg3/anes1992.dta")
anes$pidfac <- as.factor(anes$pid)
unrestricted.mod <- glm(votedem ~ retnat + pidfac + age + male +
         educ + black + south, data=anes, family=binomial)
restricted.mod <- glm(votedem ~ retnat + pid + age + male + educ +
        black + south, data=anes, family=binomial)
anova(restricted.mod, unrestricted.mod, test='Chisq')

## Analysis of Deviance Table
##
## Model 1: votedem ~ retnat + pid + age + male + educ + black + south
## Model 2: votedem ~ retnat + pidfac + age + male + educ + black + south
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1030     802.09
## 2      1025     768.00  5   34.093 2.281e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Plot of effects



Ordinal Dependent Variables

Above, we considered ordinal independent variables, but what if the dependent variable is ordered?

- There is a dependent-variable analog to what we just did for independent variables called Alternating Least Squares Optimal Scaling (ALSOS)
- Developed as a method to estimate quantitative models on qualitative data without making arbitrary and ultimately unjustifiable assumptions about category spacing.

Ordinality

Recall that ordinal means the spacing between categories is unknown.

- To the extent that a spacing between categories exists numerically (e.g., by having categories coded as increasing integers starting with one), the spacing is arbitrary and artificial.

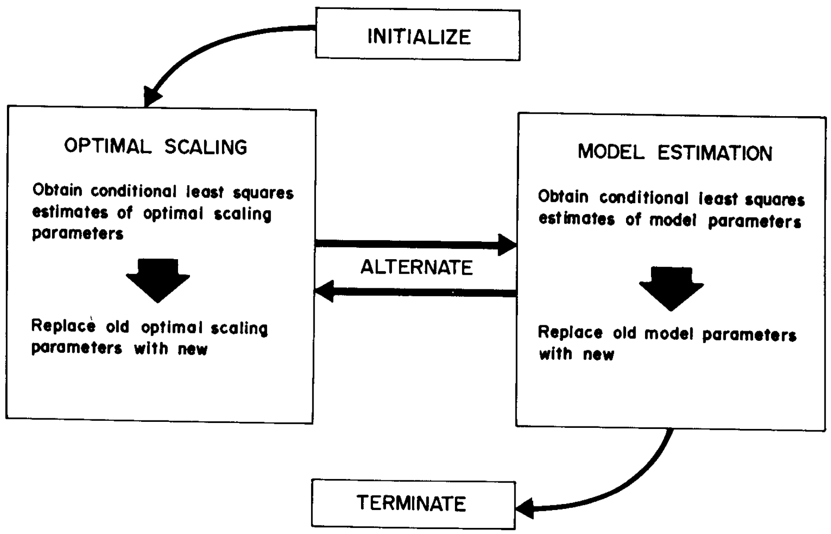
Optimal scaling can be used to assign numerical values to the categories. Bock (1960, via Young [1981]) describes optimal scaling as:

> *... a data analysis technique which assigns numerical values to observation categories in a way which maximizes the relationship between the observations and the data analysis model while respecting the measurement character of the data.*

As Young (1981) suggests:

> *If a procedure is known for obtaining a least squares description of numerical (interval or ratio measurement level) data then an ALSOS algorithm can be constructed to obtain a least squares description of qualitative data (having a variety of measurement characteristics).*

ALSOS Algorithm



In Greater Detail

Initialize algorithm by setting $\hat{y}^{(0)} = y$ and $R^{2(0)} = 0$. Then, for iterations 1:N -

1. Regress $\hat{y}^{(t-1)}$ on $\boldsymbol{X}$, save $R^{2(t)}$. If $R^{2(t)} - R^{2(t-1)} >$ tolerance, continue, otherwise end saving $\hat{y}^{(t-1)}$ as the optimally scaled values of y.
2. Optimally scale $\hat{y}^{(t)}$ against $\hat{y}^{(t-1)}$.
3. Repeat until convergence

Optimal Scaling

Assume we have the following variables on $\boldsymbol{n}$ observations:

- $\boldsymbol{o}$ (with elements o_i) which are ordered in such a way that all observations in a particular category are contiguous
- $\hat{\boldsymbol{z}}$ (with elements $\hat{z}_i$) which are model estimates in one-to-one correspondence with $\boldsymbol{o}$.
- $\boldsymbol{z}^*$ (with elements z_i^* which are optimally scaled version of $\hat{\boldsymbol{z}}$

The OS problem, then, is to find the transformation $\ell[\boldsymbol{o}] = [\boldsymbol{z}^*]$ where:

- The precise definition of $\ell[\cdot]$ depends on the measurement characteristics of $\boldsymbol{o}$, and
- $\boldsymbol{z}^*$ has a least squares relationship to $\hat{\boldsymbol{z}}$ (the model estimates of $\boldsymbol{z}^*$).

See http://forrest.psych.unc.edu/teaching/p230/LSMT-1.pdf for more on the computational details of the solution.

Measurement Level and Measurement Process

Measurement Level:

- Here, we are focusing on ordinal measurement level. We already have methods for finding optimal transformations of continuous data (to be discussed later). Though we could do this for nominal data, I think few reviewers would regard this as a viable strategy.

Measurement Process:

- Discrete: tied observations remain tied in the optimal scaling solution (Kruskal's Secondary Monotonic Transformation)

$$\ell^{do}: (o_i \sim o_m) \rightarrow (z_i^* = z_m^*)$$
$$(o_i \prec o_m) \rightarrow (z_i^* \leq z_m^*)$$

- Continuous: tied observations can become untied in the optimal scaling solution (Kruskal's Primary Monotonic Transormation)

$$\ell^{co}: (o_i \sim o_m) \rightarrow (z_i^- = z_m^-) \leq \left\{ \begin{array}{c} z_i^* \\ z_m^* \end{array} \right\} \leq (z_i^+ = z_m^+)$$
$$(o_i \prec o_m) \rightarrow (z_i^* \leq z_m^*)$$

Initialization and Convergence

- The ALSOS procedure is not guaranteed to converge to a global minimum, but to what Young (1981) calls a "conditional global optimum"
  - Where "conditional" refers to the fact that the solution is conditional on the current model parameters.
- It is possible that two different optimal scaling solutions can be arrived at by initializing the algorithm in two different ways.
  - Generally, the algorithm is initialized with least squares estimates on the raw (i.e., original) data.
  - Random starts could be chosen to assess sensitivity.

Example

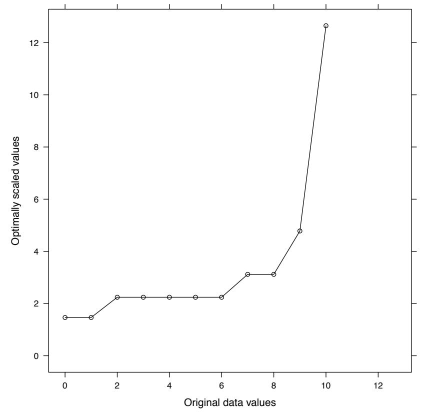
Consider Polity's Democracy variable, an 11-point scale.

- We want to know whether the spacing between polity categories as currently coded makes sense.
- Here, "makes sense" is in relation to a particular statistical model

```
library(foreign)
dat <- read.dta(
  "http://www.quantoid.net/files/reg3/linear_ex.dta")
source("http://www.quantoid.net/files/reg3/alsosdv.r")
tmp <- alsosDV(polity_dem ~ iwar + cwar + I(gdppc/10000) + logpop + rep1,
    dat, process=1, level=2, maxit=30, na.action=na.exclude, starts=NULL)
```


The Result

```
plot(tmp$result, main.title="")
```



Result: Iteration History

```
tmp$iterations

##   r-squared r-squared dif
## 1    0.3646        0.3646
## 2    0.5736        0.2089
## 3    0.5737        0.0002
```

Table: Model Comparison

	Raw	OS
(Intercept)	−1.729*	−1.692*
	(0.406)	(0.333)
iwar	1.740*	1.990*
	(0.344)	(0.282)
cwar	0.403	0.227
	(0.368)	(0.301)
I(gdppc/10000)	1.488*	2.099*
	(0.135)	(0.111)
logpop	0.483*	0.435*
	(0.045)	(0.037)
rep1	−1.347*	−1.593*
	(0.061)	(0.050)
N	2683	2683
R^2	0.365	0.574
adj. R^2	0.363	0.573
Resid. sd	3.355	2.748

Standard errors in parentheses
* indicates significance at $p < 0.05$

Sensitivity Testing

```
inits <- function(x, lower=-20, upper=20){
        tab <- table(x)
        nt <- length(tab)
        ru <- runif(nt, lower, upper)
        ru[2:nt] <- abs(ru[2:nt])
        ru <- cumsum(ru)
        newx <- ru[match(x, names(tab))]
        newx
}
res <- vector("list", 1000)
for(i in 1:1000){
res[[i]] <- alsosDV(formula, dat, maxit=30,
   na.action=na.exclude, starts=inits(dat$polity_dem,
   lower=-100, upper=100))$iterations
}
```


Diagnosing Non-Linearity

Diagnosing non-linearity in relationships between continuous predictors is a bit more tricky.

We will use an analysis of the residuals to diagnose whether the relationship between $\boldsymbol{X}$ and y is well-characterized by a line.

We will also need to figure out a flexible way to model the dependencies between $\boldsymbol{X}$ and the residuals.

- To do this, we will need to learn something about non-parametric regression

Parametric vs. Non-parametric

Our goal is to trace the dependence of y on x. Specifically, we usually want to get something like:

$$y_i | x_i = f(x_i) + e_i$$

We usually define $f(\cdot)$ to be "smooth".

- The linear functional form $(f(x_i) = \alpha + \beta x_i)$is the "smoothest" of smooth function.

The above model is parametric, because we are estimating *parameters* that describe relationship between y and x.

It is possible to characterize the relationship without estimating global parameters (i.e., parameters that apply to all of the observations equally) - what we call *non-parametric* models.

Global vs. Local Parametric Models

All of the models we will talk about below are *locally* parametric.

- They fit a parametric model to a relatively small subset of the data.
- The sum total of these many local parametric fits is a non-parametric fit - one that does not impose the same functional form for all of the data.

Because these models remain locally parametric, we can usually use information from the many local models to derive standard errors for the fit. (More on this later)

Local Polynomial Regression

To estimate the local polynomial regression between y and x, start with the smallest unique value of x (call it x_0) and you would estimate:

$$y_i w_i = \beta_0 + \beta_1 x_i w_i + \beta_2 x_i^2 w_i + \varepsilon_i w_i$$

for the *span*×100% of the observations closest to x_0. Let's say for the sake or argument that the span= 0.5.

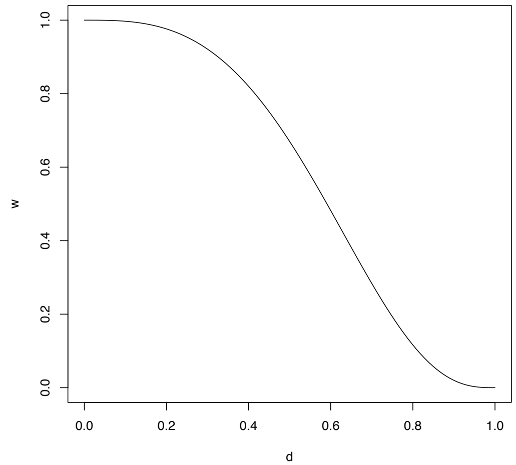
1. Find the 50% of the points closes to x_0 by calculating $d_i = |x_i - x_0|$ and then taking the 50% smallest values of d_i.
2. For the observations in the subsample, calculate the scaled distance such that $\tilde{d}_i = \frac{d_i}{max(d_i)}$. This makes the largest distance in the subsample equal to 1.
3. Calculate the weights for the subset using the tricube weight function.

$$w_i = \left(1 - \tilde{d}^3\right)^3$$

w_i for observations outside the subset will be 0.

Tricube Function

What does the tricube weighting function look like?



Robustness Weighting in Local Polynomial Regression

The steps to robustness weighted local polynomial regression are as follows:

1. Fit the local regressions using weights w_i
2. Calculate the residuals $\hat{\varepsilon}_i = y_i - \hat{y}_i$
3. Determine the median of the absolute values of the residuals $\hat{q}_{.5}$
4. Find the robustness weights (with the Bisquare weight function):

$$r_i = B\left(\frac{\hat{\varepsilon}_i}{6\hat{q}_{.5}}\right)$$

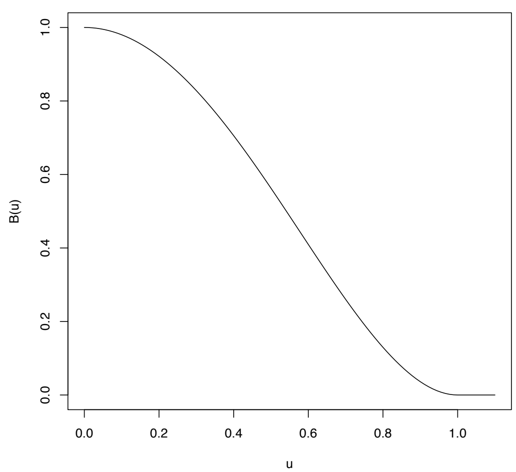
   where:

$$B(u) = \begin{cases} (1-u^2)^2, & \text{if } |u| < 1; \\ 0, & \text{otherwise.} \end{cases}$$

5. Repeat the loess procedure using weights $r_i w_i$
6. Repeat steps 2-5 until the loess model converges.

Bisquare Weighting Function

What does the bisquare weight function look like?



Choosing the Span

The choice of *span* (i.e., the number of points included in each local model) - this encapsulates the bias-variance tradeoff.

- A bigger span can induce bias which results in a non-parametric estimate that is not faithful to the local patterns in the data
- A smaller span can exhibit considerable variability while sticking very closely to the local pattern in the data. Overfitting is a potential problem here.

Overfitting is not necessarily a problem if we *only* care about the relationship in this sample. However, if we are (either explicitly or implicitly) trying to say something about a population with the sample, then overfitting can be a real problem.

Choosing Polynomial Degree and Weight Function

Polynomial Degree:

- Higher degree polynomials are more likely to overfit the data.
- The most common advice is to set the polynomial degree to 2 and adjust the span to generate the required smoothness of fit.

Weight Function:

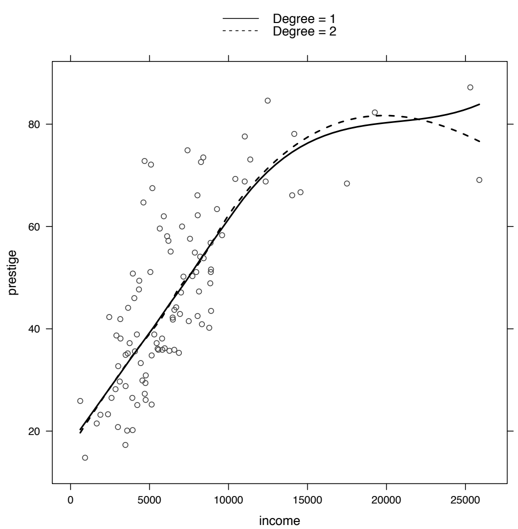
- The default in **R** is the *tricube* weight function.
- There is little reason to change this as it generally has a relatively small effect on the overall estimate.

LPR in R

There are two different versions of this type of regression: Loess and Lowess.

- In **R**, The important difference between these two is that Loess can take multiple predictors (i.e., multiple nonparametric regression) whereas Lowess only takes 1. Further, the user has much more control over `loess` than `lowess`, so we spend time on the former.
- Both `loess` and `lowess` are in the `stats` package that comes with every distribution of R.
- The robustness weighting is done by specifying `family = symmetric` in the `loess` command. Otherwise, if `family = gaussian`, no robustness weighting (only distance weighting) will be done.

Loess Graph

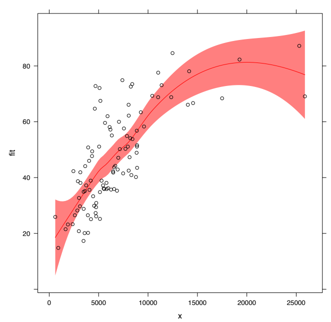


Interpretation of Non-Parametric Fits

- Often, we are tempted to impose some meaning on small bumps and dips in the local fit. As Keele (2007) suggests - “it is a temptation analysis should resist.”
- It is often useful to consider the overall general pattern in the data and if there appears to be a pattern that can be modeled parametrically - impose that fit and assess the difference between the parametric and non-parametric models (more on this later).

Plotting the LOESS curve

```
source('http://quantoid.net/files/reg3/plot.loess.r')
lo <- loess(prestige ~ income, data=Prestige, span=.75)
plot.loess(lo, addPoints=TRUE)
```



First Derivatives from LOESS Curve

Though we don't normally talk about it this way, in an OLS model, a variable's effect is the partial first derivative of the equation with respect to the variable of interest.

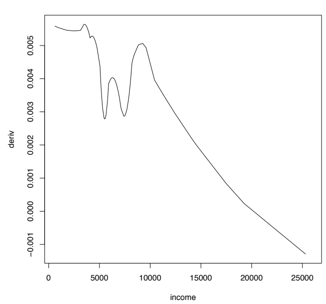
- With a parametric form, this is easily calculated.
- With non-parametric regression, we can estimate the partial first derivative with:

$$\frac{\partial f(x)}{\partial x} = \frac{f(x|x+\delta) - f(x|x)}{\delta}$$

where δ is a small number (e.g., 0.00001)

Loess Derivatives

```
source('http://quantoid.net/files/reg3/loessderiv.r')
deriv <- loessDeriv(lo)
tmp <- data.frame(income=lo$x, deriv=deriv)
tmp <- tmp[order(tmp$income), ]
plot(tmp, type="l")
```



Non-linearity

- The assumption that the average error $E(\varepsilon)$ is everywhere zero implies that the regression surface accurately reflects the dependency of Y on the X's
- We can see this as linearity in the broad sense
  - *i.e.*, non-linearity refers to a partial relationship between two variables that is not summarized by a straight line, but it could also refer to situations when two variables specified to have additive effects actually interact.
- Violating this assumption implies that the model fails to account for a systematic pattern between Y and the X's
  - Often models characterized by this violation will still provide a useful approximation of the pattern in the data, but they can also be misleading
- It is impossible to directly view the regression surface when more than two predictors are specified, but we can employ *partial residual plots* to assess non-linearity.

Partial-Residual Plots (Component-plus-residual plots)

- The partial residual for the j^{th} explanatory variable from a multiple regression is

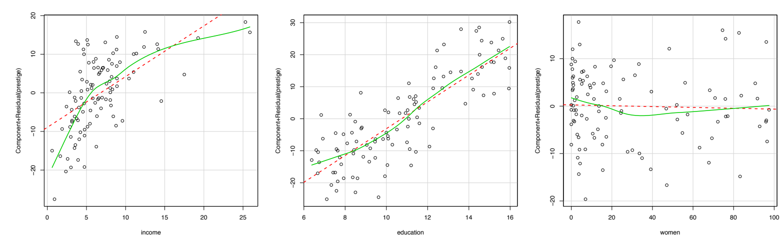
$$E_i^{(j)} = E_i + B_j X_{ij}$$

  - This simply adds the linear component of the partial regression between Y and X_j (which may be characterized by a non-linear component) to the least squares residuals
- The "partial residuals" $E^{(j)}$ are plotted versus X_j, meaning that B_j is the slope of the multiple simple regression of $E^{(j)}$ on X_j
  - A non-parametric smooth helps assess whether the linear trend adequately captures the partial relationship between Y and X.

Example of partial residual plots (1): The Canadian Prestige Data

```
data(Prestige)
Prestige$income <- Prestige$income/1000
Prestige.model<-lm(prestige ~ income + education +
        women, data=Prestige)
library(car)
crPlot(Prestige.model, "income")
crPlot(Prestige.model, "education")
crPlot(Prestige.model, "women")
```


Example of partial residual plots (2): The Canadian Prestige Data



- The plot for income suggests a power transformation down the ladder of powers; for education the departure from linearity isn't problematic; for % women, there appears to be no relationship

Testing Non-linearity with CR Plots

While this is not a substitute for looking at the graphs, I have written a couple of functions that will allow you to use an F-test to evaluate significant departures from linearity.

```
crTest(Prestige.model, adjust.method="holm")

##              RSSp   RSSnp DFnum DFdenom     F     p
## income    6033.57 4985.47 4.285  95.715 4.696 0.004
## education 6033.57 5460.73 3.034  96.966 3.352 0.043
## women     6033.57 5838.12 2.901  97.099 1.120 0.344
```


Inference for Nonparametric Models

In the example above, we are testing the local polynomial regression against the straight line in the CR Plot. The main issue is figuring out the degrees of freedom for the LPR.

We know in OLS:

- $\hat{\boldsymbol{y}} = \boldsymbol{Hy}$ and $df_{\text{model}} = tr(\boldsymbol{H})$
- $\boldsymbol{H}$ is symmetric and idempotent so $tr(\boldsymbol{H}) = tr(\boldsymbol{HH}')$
- Residual variance is $\frac{\boldsymbol{e}'\boldsymbol{e}}{tr[(\boldsymbol{I}-\boldsymbol{H})'(\boldsymbol{I}-\boldsymbol{H})]}$ where the denominator is the residual degrees of freedom.

Degrees of Freedom II

In LPR, $\boldsymbol{y} = \boldsymbol{Sy}$, we have three different degrees of freedom estimates based on the OLS properties from above:

- $tr(\boldsymbol{S})$ (df model)
- $tr(\boldsymbol{SS}')$ (df model)
- $tr[(\boldsymbol{I}-\boldsymbol{S})'(\boldsymbol{I}-\boldsymbol{S})] = n - tr(2\boldsymbol{S}+\boldsymbol{SS}')$ (df residual), so $tr(2\boldsymbol{S}+\boldsymbol{SS}'$ would be the model df.

Each provides a potentially different number with none being particularly preferred over the other.

F-Tests and Nonparametric Models

We can perform an incremental F-tests on two nonparametric models

$$F = \frac{\frac{RSS_N - RSS_A}{v_1}}{\frac{RSS_A}{\delta_1^{(A)}}}$$

where $\delta_1^{(A)}$ is as defined above for the alternative (or full) model and v_1 is $\delta_1^{(A)} - \delta_1^{(N)}$ and RSS are residual sums of squares.

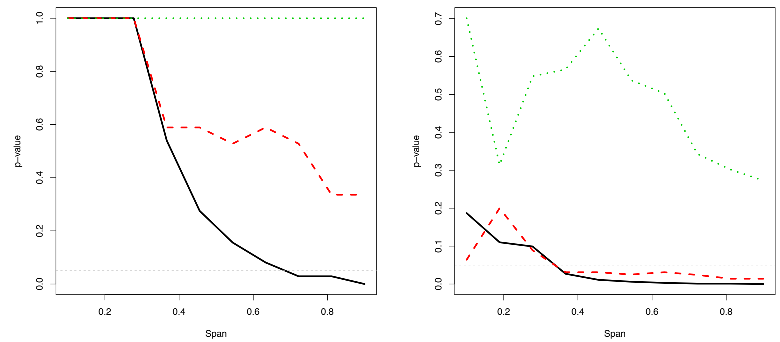
- This statistic follows and F distribution with $\frac{\left(\delta_1^{(A)}-\delta_1^{(N)}\right)^2}{\delta_2^{(A)}-\delta_2^{(N)}}$ numerator and $\frac{\delta_1^{(A)2}}{\delta_2^{(A)}}$ denominator degrees of freedom.

Different Spans in the Loess Model

```
crst <- crSpanTest(Prestige.model,
        c(.1,.9), adjust.method="holm",
        adjust.type="both")
matplot(crst$x, crst$y, type="l", lwd=3,
        xlab = "Span", ylab = "p-value")
abline(h=0.05, col="gray75", lty=2)
crst <- crSpanTest(Prestige.model,
        c(.1,.9), adjust.method="none",
        adjust.type="none")
matplot(crst$x, crst$y, type="l", lwd=3,
        xlab = "Span", ylab = "p-value")
abline(h=0.05, col="gray75", lty=2)
```


Graph from Span test

Figure: Different Spans with and without Multiple-testing Correction



(a) Holm Correction (b) No Correction

Two Dimensions of Nonlinearity

- Simple vs. Complex
  - Simple means the curvature of the function relating x to y does not change direction (i.e., there is no inflection point).
  - Complex means that there is an inflection point.
- Monotone vs Non-monotone
  - Monotone means that as x increases the function relating x to y never decreases or x increases the function relating x to y never increases, depending on the nature of the function.

Handling Non-linearity: Common Strategies

Simple, monotone

- Transformations of Y and/or X

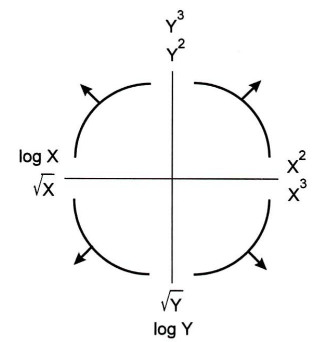
Complicated Non-linearity

1. Polynomial Regression
   - If pattern has too many turns, polynomials tend to oversmooth peaks
2. Regression Splines
3. More complicated non-parametric models.

Transformable Non-linearity: Bulging rule

- The direction of the bulge indicates the appropriate type of power transformation for Y and/or X
- A bulge to the top left of the scatterplot suggests transforming Y up the ladder of powers and/or X down the ladder of powers will straighten the relationship

Figure 4.6 from Fox (1997)



Maximum Likelihood Transformation Methods

- Although the *ad hoc* methods for assessing non-linearity are usually effective, there are more sophisticated techniques based on maximum likelihood estimation
- These techniques embed the usual multiple-regression model in a more general non-linear model that contains (a) parameter(s) for the transformation(s)
  - The transformation parameter λ is estimated simultaneously with the usual regression parameters by maximizing the likelihood and this obtaining MLEs: $\mathcal{L}(\lambda, \alpha, \beta_1, \ldots, \beta_k, \sigma_\varepsilon^2)$
  - If $\lambda = \lambda_0$ (i.e., there is no transformation), a likelihood ratio test, Wald test, or score test of $H_0 : \lambda = \lambda_0$ can assess whether the transformation is required
- If several variables need to be transformed, several such parameters need to be included

Box-Tidwell Transformation of the X's (1)

- Maximum likelihood can also be used to find an appropriate linearizing transformation for the X variables
- The Box-Tidwell model is a non-linear model that estimates transformation parameters for the X's simultaneously with the regular parameters

$$Y_i = \alpha + \beta_1 X_{i1}^{\gamma_1} + \cdots + \beta_k X_{ik}^{\gamma_k} + \varepsilon_i$$

  where the errors are iid: $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 \boldsymbol{I}_n)$ and the X_{ij} are positive
- Explicit in this model is a power transformation of each of the X's
  - Of course, we would not want to transform dummy variables and the like, so we should not attempt to estimate transformation parameters for them

Box-Tidwell Transformation of the X's (2)

The Box and Tidwell procedure yields a constructed variable diagnostic in the following way:

1. Regress Y on the X's and obtain $A, B_1, \ldots, B_k$.
2. Regress Y on the X's and the constructed variables $X_1 \log_e X_1, \ldots, X_k \log_e X_k$ to obtain $A', B_1', \ldots, B_k', D_1, \ldots, D_k$
3. The constructed variables are used to assess the need for a transformation of X_j by testing the null hypothesis $H_0 : \delta_j = 0$ where $D_j = \hat{\delta}_j$
4. A preliminary estimate of the transformation parameter γ_j is given by
$$\tilde{\gamma}_j = 1 + \frac{D_j}{B_j}$$
   where B_j is the coefficient on X_j from the original equation in step 1
5. Steps 1,2, and 4 are iterated until the transformation parameters converge

Box-Tidwell transformation Example: Prestige Data

```
data(Prestige)
boxTidwell(prestige ~ income + education,
    ~poly(women, 2), data=Prestige)

##           Score Statistic   p-value MLE of lambda
## income          -5.301289 0.0000001    -0.0377746
## education        2.405557 0.0161479     2.1928267
##
## iterations =  12
```

- A quadratic partial regression is included for women because we saw earlier that this might be needed.
- The statistically significant score tests indicate that transformations are needed for both variables
- The MLE of Power suggests that income should be transformed by a power of -0.037 (suggesting the log would work well) and education by a power of 2.19, suggesting that education^2 would suffice

Testing the Transformations

If you wanted to test whether the transformations were "close enough", you could just re-run the Box-Tidwell function on the new model with the transformed variables.

- If the transformations you provided (e.g., the log instead of -0.03) were good enough, then the transformation powers on the new data should be insignificant.

```
boxTidwell(prestige ~ log(income) + I(education^2),
    ~poly(women, 2), data=Prestige)

##                 Score Statistic   p-value MLE of lambda
## log(income)          -0.1860504 0.8524053      0.792984
## I(education^2)        0.3616705 0.7175983      1.093631
##
## iterations =  5
```

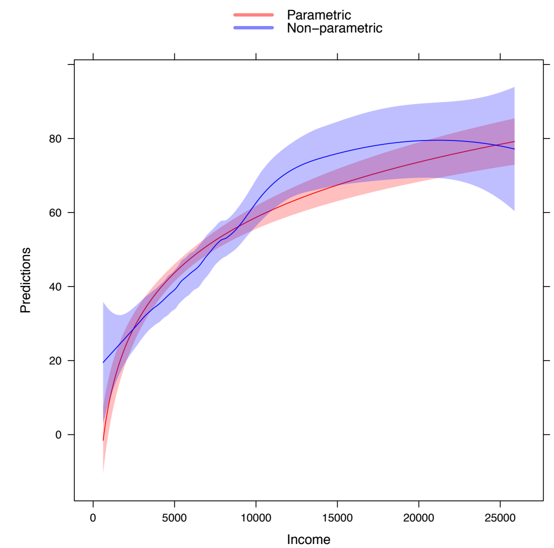
- Notice that in both cases, the p-values are > 0.05

Another Method for Testing Transformations

Another alternative would be to test the transformation against a LOESS smooth.

```
mod <- lm(prestige ~ income, data=Prestige)
trans.mod <- lm(prestige ~ log(income), data=Prestige)
loess.mod <- loess(prestige ~ income, data=Prestige, span=.5,
    family="symmetric")
testLoess(mod, loess.mod)


## F = 2.9
## Pr( > F) = 0.01
## LOESS preferred to alternative


testLoess(trans.mod, loess.mod)


## F = 1.63
## Pr( > F) = 0.14
## LOESS not statistically better than alternative
```

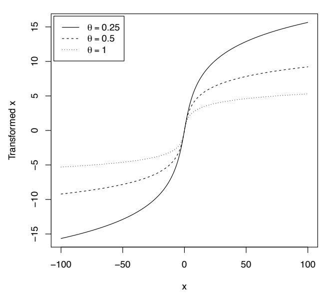

Plots of Predictions



Inverse Hyperbolic Sine Transformation

Sometimes, the log transform is not the most useful because a variable has lots of zeros and you don't want to add a constant to all counts. The IHS transformation is a good alternative.

$$\text{IHS}(x) = \frac{\sinh^{-1}(\theta x)}{\theta} = \frac{log\left(\theta x + log(\theta x^2 + 1)^{\left(\frac{1}{2}\right)}\right)}{\theta}$$



Using the IHS Transform

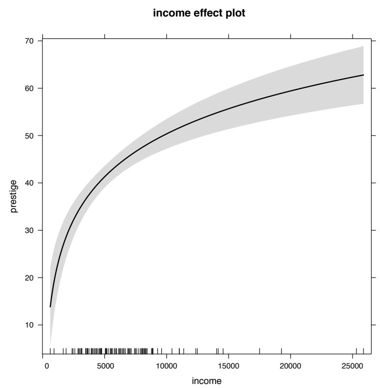
```
IHS <- function(x,theta = 1){asinh(theta*x)/theta}
trans.mod2 <- lm(prestige ~ IHS(income), data=Prestige)
summary(trans.mod2)


##
## Call:
## lm(formula = prestige ~ IHS(income), data = Prestige)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.114  -9.342  -1.218   8.101  30.454
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -154.797     18.305  -8.457 2.35e-13 ***
## IHS(income)   21.556      1.953  11.037  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.61 on 100 degrees of freedom
## Multiple R-squared:  0.5492,Adjusted R-squared:  0.5447
## F-statistic: 121.8 on 1 and 100 DF,  p-value: < 2.2e-16
```

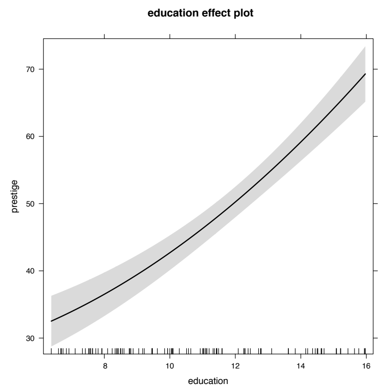
The IHS transform will also work with the `effects` package.

Effects Plots

```r
mod <- lm(prestige ~ log(income) + I(education^2) +
    poly(women, 2), data=Prestige)
plot(effect("log(income)",
        mod, default.levels=100))
```

```r
plot(effect("I(education^2)",
        mod, default.levels=100))
```





Polynomial Regression

- Two or more regressors of ascending power (i.e., linear, quadratic and cubic terms) are used to capture the effects of a single variable
  - For every bend in the curve, we add another term to the model, going up in power each time
- The terms fit a non-linear function of the explanatory variable X, but the parameters enter the formula in a linear fashion - Y is predicted by a linear combination of parameter estimates times the values of X
  - In other words, polynomial models are linear in the parameters even though they are non-linear in the variables

Order	Equation
First	$Y = \alpha + \beta_1 X$
Second	$Y = \alpha + \beta_1 X + \beta_2 X^2$
Third	$Y = \alpha + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$

Polynomial equations: How to choose the order

- It is initially useful to look at the bends in a smooth of the scatterplot or partial residual plot
  - If there is only one, a second order polynomial should be tried. For each extra bulge, we go up one in order
- A good strategy is to start with one more than you think the model needs and drop the term if it is not statistically significant
- Incremental F-tests can be used to help pick the “right” order to use in the equation
  - If the term is not statistically significant, it is usually advisable to delete the term from the model - we want as few order terms as possible
  - For orthogonal polynomials, t-tests can be used
- If the order is too high, however, the results will not be easy to interpret (higher than third order is rarely used)

Orthogonalizing Regressors

It is possible to orghotonalize the power regressors before fitting the model, below is an example for a 3^{rd} degree polynomial.

1. Create $(p_1, p_2, p_3) = (X, X^2, X^3)$
2. Use p_1 as the value for the first-degree term.
3. Regress the p_2 and p_3 on p_1 and create residuals $e_2^{(1)}$ and $e_3^{(1)}$, respectively. Use $e_2^{(1)}$ as the value for the second-degree term
4. Regress $e_3^{(1)}$ on p_1 and $e_2^{(1)}$ and use the residuals from that equation (call them $e_3^{(2)}$) as the third degree term.

This is not exactly what `poly` in R does, but the idea is similar. `poly()` also does some other normalization, so results using the above method, while equivalent in model fit terms will generate different coefficient estimates.

Orthogonal Polynomials in **R** Example: Prestige Data

- One can fit a polynomial regression by calculating the regressors individually and adding them to the regression equation - i.e., calculate and add a quadratic term X^2 and a cubic term X^3 manually.
- *Orthogonal Polynomials* can be added in a much more simple - and better - way in **R** , however, by specifying a `poly` argument to the variable. Non-orthogonal polynomials can be specified with the `raw=T` argument to `poly`.
  - The order of the polynomial is specified after the variable name

Regression Output

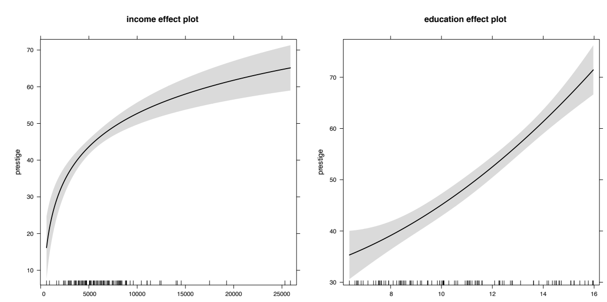
```
##
## Call:
## lm(formula = prestige ~ log(income) + poly(education, 2) + women,
##     data = Prestige)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1714  -3.7064  -0.3755   4.3029  17.2487
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -67.91272   17.08393  -3.975 0.000135 ***
## log(income)          13.07930    1.90475   6.867 6.27e-10 ***
## poly(education, 2)1 103.38447    9.63587  10.729  < 2e-16 ***
## poly(education, 2)2  12.63013    7.19449   1.756 0.082326 .
## women                 0.05082    0.02967   1.713 0.089957 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.018 on 97 degrees of freedom
## Multiple R-squared:  0.8402,Adjusted R-squared:  0.8336
## F-statistic: 127.5 on 4 and 97 DF,  p-value: < 2.2e-16
```

- Since orthogonal polynomials were used, the t-test for the individual parameters is all that is needed. An F-test will show nothing different
- Nonlinear effects are difficult to comprehend in numerical form. Graphing the fitted values provides a much better alternative.

Effect Displays for Income and Education

```
library(effects)
plot(effect("log(income)", mod,
    default.levels=100, se=T))
```

```
plot(effect("poly(education, 2)", mod,
    default.levels=100, se=T))
```



Readings

Today: Linearity Diagnostics

∗ Fox (2008) Chapters 4, 12 (Sections 12.3-12.5) & 17
∗ Fox (2002) Chapter 3
∗ Jacoby (1999)
– Fox (2000)

Fox, John. 2000. *Nonparametric Simple Regression.* Thousand Oaks: Sage.

Fox, John. 2002. *An R and S-Plus Companion to Applied Regression.* Thousand Oaks: Sage Publications.

Fox, John. 2008. *Applied Regression Analysis and Generalized Linear Models, 2^{nd} edition.* Thousand Oaks, CA: Sage, Inc.

Jacoby, William G. 1999. "Levels of Measurement and Political Research: An Optimistic View." *American Journal of Political Science* 43(1):271–301.