Regression III

Linearity I

Dave Armstrong

Outline for Linearity Discussion

- The linearity assumption
- Diagnosis of un-modeled non-linearity (CR Plots, Smoothers)
- Simple remedies for un-modeled non-linearity (transformations, polynomials).
- 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:

- Characterize the relationship between y and x with a single (or small set of) numbers.

- Easily interpret the marginal effect of x.

- Easily present the results of the modeling enterprise.

Notes

Type notes here...

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.

Notes

Type notes here...

Linearity and Multi-Category Variables

Nominal Variables $\rightarrow$ Dummy Regressors $\rightarrow$ 😄

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*

Notes

Type notes here...

Testing the Hypothesis

Consider the model (Covariates can be added to the model below without loss of generality):

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

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

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

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.

Notes

Type notes here...

Testing the Hypothesis: Ordinal Variables

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

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

where $I()$ is an indicator function such that $I(\cdot)$ is 1 if the expression inside is true and 0 otherwise.

Notes

Type notes here...

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?

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

Notes

Type notes here...

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:

- Run the model by creating dummy variables for all but the smallest category of the variable in question.

- Test the appropriate restrictions on the model.

Notes

Type notes here...

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
```

Notes

Type notes here...

Hypothesis Test

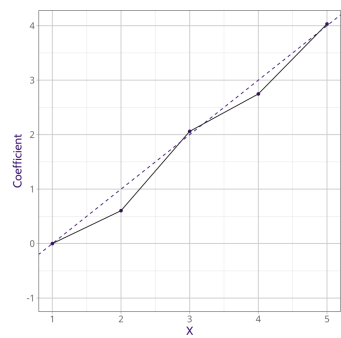We can also perform a hypothesis test using the general linear hypothesis testing:

```
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
```

Notes

Type notes here...

Linear vs. Non-linear effect

Notes

Type notes here...

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
```

Notes

Type notes here...

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
```

Notes

Type notes here...

Plot of effects
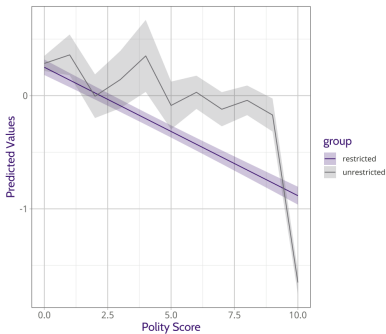
Notes

Type notes here...

Linearity of Factors in GLMs

```r
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
```

Notes

Type notes here...

Plot of effects
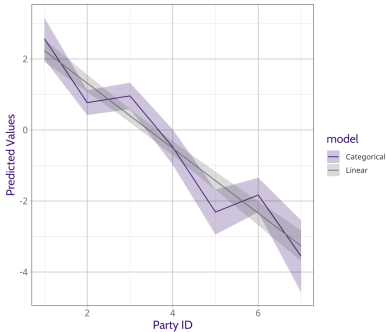
Notes

Type notes here...

Monotonicity

Sometimes we may want to consider only a subset of possible alternative forms.

- With ordinal variables, if relationships are monotonic, they might very well be consistent with our hypotheses even if they are not linear.
- Testing whether a monotonic (though perhaps not linear) model is not significantly worse than a fully unconstrained model is a nice "middle-ground".

33 / 77

Notes

Type notes here...

Example 1

```
set.seed(519)
x <- rep(1:5, 100)
x <- x[order(x)]
means <- c(0, .25, .5, .45, 1)
y <- 2 + means[x] + rnorm(500,0,1)
x <- as.factor(x)
df <- data.frame(y=y, x=x)
m1 <- lm(y ~ x, data=df)
```

```
summary(m1)

##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.94925 -0.64909 -0.01495  0.66791  2.85968
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0532     0.0983  20.887  < 2e-16 ***
## x2            0.1164     0.1390   0.837 0.402747
## x3            0.5169     0.1390   3.718 0.000224 ***
## x4            0.3678     0.1390   2.646 0.008405 **
## x5            0.9019     0.1390   6.487 2.12e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.983 on 495 degrees of freedom
## Multiple R-squared:  0.09551,    Adjusted R-squared:  0.0882
## F-statistic: 13.07 on 4 and 495 DF,  p-value: 3.994e-10
```

Notes

Type notes here...

The test

```
library(ggeffects)
e1 <- ggpredict(m1,)
plot(e1)
```

```
## $x
```

Predicted values of y

```
df <- df %>% mutate(
    x_num = as.numeric(x),
    x_mono = case_when(
        x_num == 4 ~ 3, TRUE ~ x_num),
    x_mono = factor(x_mono, levels=c(1,2,3,5),
                    labels=c("1", "2", "3-4", "5"))
    )
m2 <- lm(y ~ x_mono, data=df)
anova(m1, m2, test="F")
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x
## Model 2: y ~ x_mono
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    495 478.32
## 2    496 479.43 -1   -1.1106 1.1493 0.2842
```

```
car::linearHypothesis(m1,
                "x3=x4")
```

```
## Linear hypothesis test
##
## Hypothesis:
## x3 - x4 = 0
##
## Model 1: restricted model
## Model 2: y ~ x
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    496 479.43
## 2    495 478.32  1    1.1106 1.1493 0.2842
```

Notes

Type notes here...

Automatic Inference

Sometimes, it is not obvious what is the best way to change the pattern to make it monotonic.

```
library(ic.infer)
mon.x <- make.mon.ui(df$x)
mon.mod <- orlm(m1,ui=mon.x, index=2:5)
```

```
Order-restricted linear model with restrictions of coefficients of
x2 x3 x4 x5

 Inequality restrictions:
     x2 x3 x4 x5
1:   1  0  0  0  %*%colnames >= 0
2:  -1  1  0  0  %*%colnames >= 0
3: A 0  -1 1  0  %*%colnames >= 0
4:   0  0  -1 1  %*%colnames >= 0

 Note: Restrictions marked with A are active.
```

```
summary(mon.mod, brief=TRUE)
```

```
## Order-restricted linear model with restrictions of coefficients of
## x2 x3 x4 x5
##
##
## Coefficients from order-restricted model:
##   (Intercept)          R x2          R x3          R x4          R x5
##     2.0532369     0.1164204     0.4423570     0.4423570     0.9018621
##
##  Note: Coefficients marked with R are involved in restrictions.
##
## Hypothesis tests ( 495 error degrees of freedom ):
## Overall model test under the order restrictions:
##       Test statistic: 0.09360797,  p-value: <0.0001
##
## Type 1 test: H0: all restrictions active(=)
##        vs. H1: at least one restriction strictly true (>)
##       Test statistic: 0.09360797,  p-value: <0.0001
## Type 2 test: H0: all restrictions true
##        vs. H1: at least one restriction false
##       Test statistic: 0.002316513,   p-value: 0.6841
## Type 3 test: H0: at least one restriction false or active (=)
##        vs. H1: all restrictions strictly true (>)
##       Test statistic: -1.072071,   p-value: 0.8579
##
## Type 3 test based on t-distribution (one-sided),
## all other tests based on mixture of beta distributions
```

Notes

Type notes here...

Polity Example

Testing fully constrained (linear) against fully unconstrained (factor) model:

```
unrestricted.mod <- lm(rep1 ~ polity_dem_fac + iwar +
    cwar + logpop + gdppc,data=dat)
restricted.mod <- lm(rep1 ~ polity_dem + 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
```

Type notes here...

```
mon.polity <- make.mon.ui(dat$polity_dem_fac)*-1
mon.mod <- orlm(unrestricted.mod,ui=mon.polity,
                index=2:11)
summary(mon.mod, brief = TRUE)
```

```
## Order-restricted linear model with restrictions of coefficients of
## polity_dem_fac1 polity_dem_fac2 polity_dem_fac3 polity_dem_fac4 polity_dem_fac5 polity_dem_fac6 polity_dem_fac7 polity_dem_fac8 polity_dem_fac9
##
## Coefficients from order-restricted model:
##        (Intercept)  R polity_dem_fac1  R polity_dem_fac2  R polity_dem_fac3
##        -1.649968068       0.000000000       -0.183968744       -0.183968744
##   R polity_dem_fac4  R polity_dem_fac5  R polity_dem_fac6  R polity_dem_fac7
##        -0.183968744      -0.297738444      -0.297738444       -0.365601155
##   R polity_dem_fac8  R polity_dem_fac9 R polity_dem_fac10            iwar
##        -0.365601155      -0.461183725      -1.937221990       0.893868406
##              cwar            logpop            gdppc
##        0.657546894       0.232332361      -0.000064166
##
##  Note: Coefficients marked with R are involved in restrictions.
##
## Hypothesis tests ( 2668 error degrees of freedom ):
## Overall model test under the order restrictions:
##      Test statistic: 0.7016129,   p-value: <0.0001
##
## Type 1 test: H0: all restrictions active(=)
##          vs. H1: at least one restriction strictly true (>)
##      Test statistic: 0.2768493,   p-value: <0.0001
## Type 2 test: H0: all restrictions true
##          vs. H1: at least one restriction false
##      Test statistic: 0.002170722,   p-value: 0.7097
## Type 3 test: H0: at least one restriction false or active (=)
##          vs. H1: all restrictions strictly true (>)
##      Test statistic: -1.026567,   p-value: 0.8476
##
```

Type notes here...

Monotonicity in GLMs

The full suite of functions for testing is not available for non-Gaussian GLMs. Could do the following:

1. Estimate the unrestricted (factor) GLM.
2. Save the fitted response $\hat{\eta}$.
3. Regress $g(\hat{\eta})$ on the monotone restrictions, where $g(\cdot)$ is the link function.
4. Use the results impose the appropriate monotonicity constraints on the variable of interest.
5. Re-estimate the GLM with the imposed monotonicity restrictions and test against the unconstrained model.

Notes

Type notes here...

Example

- Estimate Model:

```
anes_tmp <- anes %>%
  dplyr::select(votedem, retnat, pidfac, age, male, educ, black, south, pid) %>%
  na.omit()
unrestricted.mod <- glm(votedem ~ retnat + pidfac + age + male +
    educ + black + south, data=anes_tmp, family=binomial)
```

- Save $\hat{\eta}$ to use later.

```
anes_tmp <- anes_tmp %>%
  mutate(eta_hat = predict(unrestricted.mod,
                           type="response"))
```

- Regress $g(\hat{\eta})$ on the monotone restrictions.

```
mon.anes <- make.mon.ui(anes_tmp$pidfac)*-1
tmp_mod <- lm(qlogis(eta_hat) ~ retnat + pidfac +
    age + male + educ + black + south, data=anes_tmp)
mon.mod <- orlm(tmp_mod, ui=mon.anes, index=4:9)
mon.mod
```

Notes

Type notes here...

Example continued

- Use results from `orlm` to impose appropriate restrictions

```
anes_tmp <- anes_tmp %>%
  mutate(pidfac2 = case_when(
    pid == 1 ~ 1,
    pid %in% c(2,3) ~ 2,
    pid ==4 ~ 3,
    pid %in% c(5:6) ~ 4,
    pid == 7 ~ 5),
    pidfac2 = as.factor(pidfac2))
```

- Re-estimate model and test against unconstrained model.

```
mon.glm <- glm(votedem ~ retnat + pidfac2 + age + male +
    educ + black + south, data=anes_tmp, family=binomial)
umod <- update(unrestricted.mod, data=anes_tmp)
anova(mon.glm, umod, test='Chisq')

## Analysis of Deviance Table
##
## Model 1: votedem ~ retnat + pidfac2 + age + male + educ + black + south
## Model 2: votedem ~ retnat + pidfac + age + male + educ + black + south
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1027     770.03
## 2      1025     768.00  2    2.026   0.3631
```

Notes

Type notes here...

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.

Notes

Type notes here...

Ordinality

Recall that ordinal means the spacing between categories is unknown.

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).

Notes

Type notes here...

ALSOS Algorithm

Notes

Type notes here...

In Greater Detail

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

- Regress $\hat{y}^{(t-1)}$ on $\mathbf{X}$, save $R^{2(t)}$. If $R^{2(t)} - R^{2(t-1)} > \text{tolerance}$, continue, otherwise end saving $\hat{y}^{(t-1)}$ as the optimally scaled values of y.

- Optimally scale $\hat{y}^{(t)}$ against $\hat{y}^{(t-1)}$.

- Repeat until convergence

Notes

Type notes here...

Optimal Scaling

Assume we have the following variables on n observations:

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

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

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

Notes

Type notes here...

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.

Notes

Type notes here...

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)

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

Notes

Type notes here...

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.

Notes

Type notes here...

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(DAMisc)
library(foreign)
dat <- read.dta(
  "http://www.quantoid.net/files/reg3/linear_ex.dta")
tmp <- alsosDV(polity_dem ~ iwar + cwar + I(gdppc/1000) + logpop + rep1,
    dat, process=1, level=2, maxit=30, na.action=na.exclude, starts=NULL)
```

Notes

Type notes here...

The Result

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

Optimal Transformation for Variable: orig_data[[dv]]
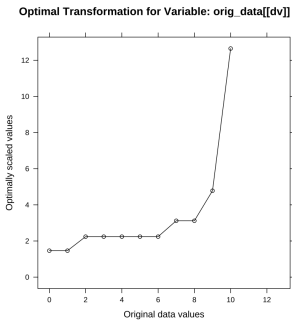
Notes

Type notes here...

Result: Iteration History

```
tmp$iterations
```

```
##    r-squared r-squared dif
## 0    0.3646        0.3646
## 1    0.5736        0.2089
## 2    0.5737        0.0002
```

Notes

Type notes here...

Models

```
##                              ##
## ======================================================
##                        Dependent variable:
##                   ----------------------------------
##                   polity_dem   polity_dem_os
##                      (1)            (2)
## ------------------------------------------------------
## 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)
##
## Constant          -1.729***      -1.692***
##                    (0.406)        (0.333)
##
## ------------------------------------------------------
## Observations       2,683          2,683
## R2                 0.365          0.574
## Adjusted R2        0.363          0.573
## Residual Std. Error (df = 2677)  3.355    2.748
```

Notes

Type notes here...

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
}
```

Notes

Type notes here...