Making the Most of Regression

POST 8000 – Foundations of Social Science Research for Public Policy

Steven V. Miller

Department of Political Science

Goal for Today

Discuss tricks to improve the information you can extract from a regression model.

Making the Most of Regression

Regression modeling is also storytelling.

- Your audience is not going to 100% understand what you're doing.
- That won't stop them from asking questions.

There's something you need to tell them from your model.

- *Make sure you tell them!*

Think of this like "Chekhov's gun."

- If it's in your model, be prepared to explain it.
- If you have to explain it, make it as intuitive as possible.

Two Quickies

Here are two recurring things a lay audience will ask:

1. What is the “constant” or “y-intercept?”
2. Do bigger coefficients mean bigger effects relative to other coefficients?

The Problem of the Constant

You know by now the "constant" or "y-intercept" is not a coefficient.

- It's just an estimate of y when all xs are set to 0.

Your audience is going to want to interpret it.

- Worse yet: your audience is going to want to interpret something that probably makes no sense as you've been doing it.
- 0 is typically not an available or plausible response in raw regression inputs.

What's a Bigger Effect?

Your coefficients will rarely, if ever, share a common scale.

- Absent a common scale, comparing regression coefficients is a fool's exercise.
- Coefficients are in part a function of the scale.
    - i.e. binary IVs will typically have larger coefficients (saying nothing of significance).

You and your audience will want to compare regression coefficients.

- However, your presentation will probably preclude this.

A Simple Illustration

Let's illustrate this with a simple data set in `post8000r`.

```r
EASV16 %>% select(state, percoled, gdp16, sunempr12md, trumpshare)
```

```
## # A tibble: 52 x 5
##    state                percoled    gdp16 sunempr12md trumpshare
##    <chr>                   <dbl>    <dbl>       <dbl>      <dbl>
##  1 Alabama                  24.7  203830.      -0.2        62.1
##  2 Alaska                   29.6   49363.       0.3        51.3
##  3 Arizona                  28.9  311091       -0.600      48.7
##  4 Arkansas                 22.4  120375.      -0.6        60.6
##  5 California               32.9 2657798.      -0.300      31.6
##  6 Colorado                 39.9  329368.      -0.6        43.3
##  7 Connecticut              38.6  263696.      -0.700      40.9
##  8 Delaware                 31     69550.      -0.2        41.9
##  9 District of Columbia     56.8  129826.      -0.5         4.07
## 10 Florida                  28.6  938774.      -0.400      49.0
## # ... with 42 more rows
```

A Simple Illustration

Let's see if we can model the share of the vote Trump received (`trumpshare`) with four variables:

- the percent of the state (25 and older) with a college diploma (`percoled`).
- the GDP of the state in 2016 (`gdp16`).
- Whether the state is in the South (`south`).
- The 12-month difference in the state unemployment rate for Nov. 2016 (`sunempr12md`).
    - Higher values = rising unemployment relative to point from Nov. 2015.

Note: this is clearly a simple exercise and we'll ignore some other model problems.

- e.g. DC is a huge outlier, which is why we'll ignore it.
- Probably should logarithmically transform the GDP variables too.

The Statistical Model

```r
EASV16 %>%
    mutate(south = ifelse(region == "South", 1, 0)) -> EASV16
M1 <- lm(trumpshare ~ percoled + sunempr12md + gdp16 + south,
         subset(EASV16, state != "District of Columbia"))
broom::tidy(M1) %>%
    kable(.)
```

term	estimate	std.error	statistic	p.value
(Intercept)	88.8449574	6.3246455	14.047421	0.0000000
percoled	-1.2017423	0.2017189	-5.957511	0.0000004
sunempr12md	7.9977018	2.5329840	3.157423	0.0028404
gdp16	-0.0000056	0.0000020	-2.802234	0.0074587
south	3.2570357	2.1140976	1.540627	0.1304109

Describing This Model

Here's how you would describe what you see here:

- A one-unit increase in % of the state with a college diploma decreases the Trump vote share by -1.20 percentage points.
- A one-unit increase in the state's GDP decreases Trump's vote share by -.000005 points.
- A one-unit increase in the state unemployment change increases Trump's vote share by 7.99 points.
- Being in the South increases Trump's vote share by 3.25 points.
- All but the South dummy are statistically significant (albeit one at a lower threshold).

However, there are several unsatisfying things about this model output.

The Problem of the Constant

To start: the intercept suggests Trump's vote share is expected to be 88.84% in a state where:

- No one graduated from college, AND
- There was no change from Nov. 2015 in the unemployment rate, AND
- The state isn't in the South, AND
- the GDP of the state is zero.

Constants/y-intercepts come standard in model output, but this parameter is useless as it is.

- You won't ever observe a case like this.

The Problem of the Coefficients

What's the biggest effect, as a magnitude? You won't know.

- `percoled` is the most precise, but that's not magnitude.

All variables work on a different scale.

- `percoled` has a minimum of 20.8 (WV) and a maximum of 42.7 (MA).
- `gdp16` has a minimum of 31,659 (VT) and a maximum of 2,657,798 (CA).
- `sunempr12md` has a minimum of -1 (NV) and a maximum of .4 (OH).
- `south` is a dummy/fixed effect and can only be 0 or 1.

A Solution: Scaling (by Two Standard Deviations)

Statisticians recommend scaling your *non-binary* IVs, but Gelman (2008) has a better idea: **scale by two standard deviations instead of just one.**

- The transformed variable will have a mean of 0 and a SD of .5.
- Regression coefficient would communicate estimated change in y for change across ~47.7% of data in x.

Sometimes this is what you want to communicate:

- i.e. do you really care about the effect of age going from 20 to 21? Or 50 to 51?
- Don't you want something larger/more substantive to communicate across the range of the data?

The Added Benefit of Scaling by Two Standard Deviations

Gelman (2008) notes that scaling by 2 SDs instead of 1 puts non-binary IVs on a (roughly) common scale with binary IVs.

- Assume a dummy IV d with a 50/50 split. Then: $p(d = 1) = .5$.
- Then, the SD equals .5 ($\sqrt{.5 * .5} = \sqrt{.25} = .5$)
- We can directly compare this dummy variable with our new standardized input variable!

This works well in most cases, except when $p(d = 1)$ is really small.

- e.g. $p(d = 1) = .25$, then $\sqrt{.25 * .75} = .4330127$

How to Scale by Two Standard Deviations

The process looks similar to how to calculate a z-score (with the obvious change in the denominator).

- `rescale()` in the `arm` package will do it.
- `r2sd()` in my `stevemisc` package will do it.

You could also do it manually.

```
rescale <- function(x) { (x - mean(x, na.rm=T))/(2*sd(x, na.rm=T)) }
```

How to Scale by Two Standard Deviations

```
EASV16 %>%
    # we'll use the custom function we just wrote in this document.
    mutate_at(vars("percoled", "gdp16","sunempr12md"),
              list(z = ~rescale(.))) %>%
    rename_at(vars(contains("_z")),
              ~paste("z", gsub("_z", "", .), sep = "_") ) -> EASV16

# Observe
mean(EASV16$z_percoled)
```

```
## [1] 1.185419e-16
```

```
sd(EASV16$z_percoled)
```

```
## [1] 0.5
```

An Improved Statistical Model

```
M2 <- lm(trumpshare ~ z_percoled + z_sunempr12md + z_gdp16 +
             south, subset(EASV16, state != "District of Columbia"))
broom::tidy(M2) %>%
    kable(.)
```

term	estimate	std.error	statistic	p.value
(Intercept)	47.657768	1.104007	43.167978	0.0000000
z_percoled	-14.937893	2.507405	-5.957511	0.0000004
z_sunempr12md	5.779390	1.830414	3.157423	0.0028404
z_gdp16	-5.230100	1.866404	-2.802234	0.0074587
south	3.257036	2.114098	1.540627	0.1304109

Interpreting an Improved Statistical Model

- The typical state not in the South had an estimated Trump vote share of 47.65%. The intercept is much more informative.
  - Caveat: states are weighted equally, even as there are more Californians than South Carolinians.
  - i.e. you could've added Trump's SC votes to his CA tally and he'd still lose CA by >3 million votes.
- The education variable looks to have the largest effect when everything is on a mostly common scale.
  - i.e. the effect of going from, say, a standard deviation below the mean to a standard deviation above the mean is an estimated decrease in Trump's vote share by 14.93 points.
- The unemployment variable and the GDP variable look to have similar magnitude effects (albeit in absolute terms).

Interpreting an Improved Statistical Model

Notice what didn't change.

- Scaling the other variables doesn't change the binary IVs.
- The *t*-statistics don't change either even as the coefficients and standard errors change.

Readable Regression Tables

Remember: your analysis should be as easily interpretable as possible.

- I should get a preliminary glimpse of effect size from a regression.
- Your y-intercept should be meaningful.

Standardizing variables helps.

- Creates meaningful zeroes (i.e. the mean).
- Coefficients communicate magnitude changes in x.
- Standardizing by two SDs allows for easy comparison with binary predictors.

Satisfy Your Audience

You need to relate your analysis to both me and your grandma.

- I will obviously know/care more about technical details.
- Grandma may not, but she may be a more important audience than me.

Her inquiries will probably be understandable. Examples from the above analysis:

- What's Trump's expected vote share in a better-educated Southern state?
- What's Trump's expected vote share in a better-educated state whose unemployment rate increased?

These are perfectly reasonable questions to ask of your analysis.

- If your presentation isn't prepared to answer her questions, you're not doing your job.

Statistical Presentations

Statistical presentations should:

1. Convey precise estimates of quantities of interest.
2. Include reasonable estimates of *uncertainty* around those estimates.
3. Require little specialized knowledge to understand Nos. 1 and 2.
4. Not bombard the audience with superfluous information.

We will do this with post-estimation simulation using draws from a multivariate normal distribution (King et al. 2000).

Estimating Uncertainty with Simulation

Any statistical model has a stochastic and systematic component.

- **Stochastic**: $Y_i \sim f(y_i \,|\, \theta_i, \alpha)$
- **Systematic**: $\theta_i = g(x_i, \beta)$

For a simple OLS model (i.e. a linear regression):

$$\begin{aligned} Y_i &= N(\mu_i, \, \sigma^2) \\ \mu_i &= X_i \beta \end{aligned}$$

Understanding our Uncertainty

We have two types of uncertainty.

1. **Estimation uncertainty**
   - Represents systematic components; can be reduced by increasing sample size.
2. **Fundamental uncertainty**
   - Represents stochastic component; exists no matter what (but can be modeled).

Getting our Parameter Vector

We want a **simulated parameter vector**, denoted as:

$$\hat{\gamma} \sim vec(\hat{\beta}, \hat{\alpha})$$

Central limit theorem says with a large enough sample and bounded variance:

$$\tilde{\gamma} \sim N(\hat{\gamma}, \hat{V}(\hat{\gamma}))$$

In other words: distribution of quantities of interest will follow a multivariate normal distribution with mean equal to $\hat{\gamma}$, the simulated parameter vector.

Getting our Quantities of Interest

This is a mouthful! Let's break the process down step-by-step.

1. Run your regression. Get your results.
2. Choose values of explanatory variable (as you see fit).
3. Obtain simulated parameter vector from estimating systematic component.
4. Simulate the outcome by taking random draw from the stochastic component.

Do this m times (typically $m = 1000$) to estimate the full probability distribution of your quantity of interest.

How Do You Do This?

There are a variety of packages that can do this:

- `Zelig` is primarily responsible for this movement.
- `sim()` in the `arm` package can make simulations from a multivariate normal distribution.
- `tidybayes` is a go-to for Bayesian approaches (which give you these for free anyway).
    - That's next week, though.

My approach leans on `arm::sim()`.

- This is the foundation for my `get_sims()` function in `stevemisc`.
    - However, I wrote that for mixed effects models and will tweak it soon for you.

A Quantity of Interest

What's Trump's expected vote share in a better-educated Southern state?

- What about relative to non-Southern states?

Here's how to approach doing this.

- Caveat: this is obviously a low-powered cross-sectional, observational data set of a prominent event where proper nouns matter.
- But: the approach is still informative for a wide variety of applications.

Create a New Data Frame

First, create a new data frame that match these parameters.

```
newdat = data.frame(z_percoled = c(0,0, 1,1),
                    south = c(0,1,0,1),
                    z_sunempr12md = 0,
                    z_gdp16 = 0,
                    trumpshare = 0)
```

Create a New Data Frame

```
newdat
```

```
##   z_percoled south z_sunempr12md z_gdp16 trumpshare
## 1          0     0             0       0          0
## 2          0     1             0       0          0
## 3          1     0             0       0          0
## 4          1     1             0       0          0
```

Create a Model Matrix from the Model and this new data.frame

```
MM <- model.matrix(terms(M2),newdat)
```

Get Simulations of Model Parameters via arm::sim()

```r
set.seed(8675309) # Jenny, I got your number
simM2 <- arm::sim(M2, n.sims = 1000)
```

Here's a Glimpse of These Simulated Betas

```r
as_tibble(coef(simM2)[1:5,])
```

```
## # A tibble: 5 x 5
##   `(Intercept)` z_percoled z_sunempr12md z_gdp16  south
##           <dbl>      <dbl>         <dbl>   <dbl>  <dbl>
## 1          48.0      -17.8          7.16   -5.55  2.36
## 2          48.9      -15.1          3.16   -7.02  0.498
## 3          48.3      -16.9          6.64   -5.61  0.510
## 4          50.2      -15.9          4.49   -3.61 -1.61
## 5          48.9      -16.3          6.51   -4.60  2.68
```

Calculate/Store Quantities of Interest

```
# Create blank Sims object
Sims <- tibble(y = numeric(),
               sim = numeric())

# For the 1,000 sims we have...
for(i in (1:1000)) {
  hold_me <- NULL # blank hold_me object
  # matrix multiplication from our model matrix with simulated coefs
  yi <- MM %*% coef(simM2)[i,]
  # note which of the 1,000 simulations it is.
  sim <-rep(i, length (yi))
  # cbind the QIs with the simulation indicator
  hold_me <- as_tibble(cbind(yi, sim)) %>% rename(y = V1) #
  # Bind the simulations together...
  Sims <- bind_rows(Sims, hold_me)
}
```

Calculate/Store Quantities of Interest

```
Sims
```

```
## # A tibble: 4,000 x 2
##          y   sim
##      <dbl> <dbl>
##  1    48.0     1
##  2    50.4     1
##  3    30.2     1
##  4    32.6     1
##  5    48.9     2
##  6    49.4     2
##  7    33.8     2
##  8    34.3     2
##  9    48.3     3
## 10    48.8     3
## # ... with 3,990 more rows
```

Remember What You're Looking At

Next, take inventory of what you're looking at based on the `newdat` object.

```
newdat %>%
    # replicate newdat 1000 times for our 1000 sims
    slice(rep(row_number(), 1000)) %>%
    # bind_col
    bind_cols(Sims, .) -> Sims
```

Summarize As You See Fit

The world is your oyster when you do these post-estimation simulations.

```r
Sims %>%
    group_by(south, z_percoled) %>%
    summarize(mean = mean(y),
              lwr = quantile(y, .025),
              upr = quantile(y, .975))
```

```
## # A tibble: 4 x 5
## # Groups:   south [2]
##   south z_percoled  mean   lwr   upr
##   <dbl>      <dbl> <dbl> <dbl> <dbl>
## 1     0          0  47.8  45.6  50.1
## 2     0          1  32.8  27.6  38.2
## 3     1          0  50.9  47.1  54.6
## 4     1          1  35.9  28.7  43.5
```

Conclusion

Regression provides all-else-equal effect sizes across the range of the data.

- You can extract meaningful quantities of interest from regression output itself.
- Typically, you'll need more to answer substantive questions and provide meaningful quantities of interest.
- You can help yourself by scaling your non-binary inputs by two SDs.

Post-estimation simulation from a multivariate normal distribution does this.

- When you start doing this yourselves, be prepared to provide quantities of interest for your audience.
- Never forget: *you're trying to tell a story.* Tell it well.

Table of Contents