Logistic Regression

POST 8000 – Foundations of Social Science Research for Public Policy

Steven V. Miller

Department of Political Science

Goal for Today

Discuss logistic regression, perhaps the most common form of regression.

OLS

OLS has a ton of nice properties.

- Best linear unbiased estimator (BLUE)
- Simple to execute and interpret.

It'd be a *shame* if something were to happen to one of your assumptions.

The Problem of Binary DVs

The biggest problem you'll encounter will concern your DV.

- OLS assumes the DV is distributed normally.

You'll most often encounter DVs that are binary.

- Candidate won/lost.
- Citizen voted/did not vote.
- Program succeeded/failed.
- War happened/did not happen.

Most social/political phenomena are typically "there"/"not there."

The Problem of Binary DVs

Observe this simple data frame, D.

```
D
```

```
## # A tibble: 10,000 x 2
##         x     y
##     <dbl> <int>
##  1      0     0
##  2      0     0
##  3      0     0
##  4      0     0
##  5      0     0
##  6      0     0
##  7      0     1
##  8      0     0
##  9      0     0
## 10      0     0
## # ... with 9,990 more rows
```

The Problem of Binary DVs

This simple `D` data frame is simulated where:

- `x` is a five-item ordered categorical variable [0:4].
- `y` is a binary variable with only 0s and 1s.
- The effect of a one-unit increase of `x` on `y` is 1.4.
- `y` is estimated to be -2.8 when `x` is 0.

Importantly: responses are simulated from a binomial distribution.

- Seed is set for 100% reproducibility.

The Problem of Binary DVs

Here's what OLS produces.

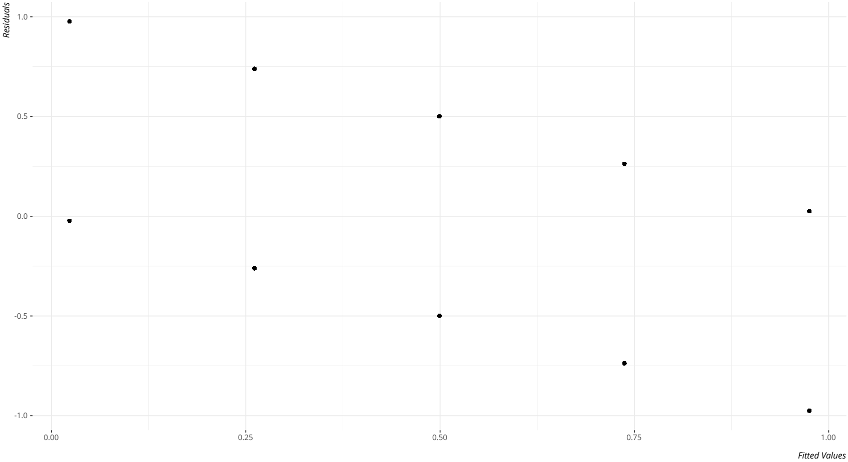
```r
M1 <- lm(y ~ x, D)
broom::tidy(M1) %>%
    mutate_if(is.numeric, ~round(., 2)) %>%
    kable(., "markdown")
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.02	0.01	3.62	0
x	0.24	0.00	91.02	0

Not even close.

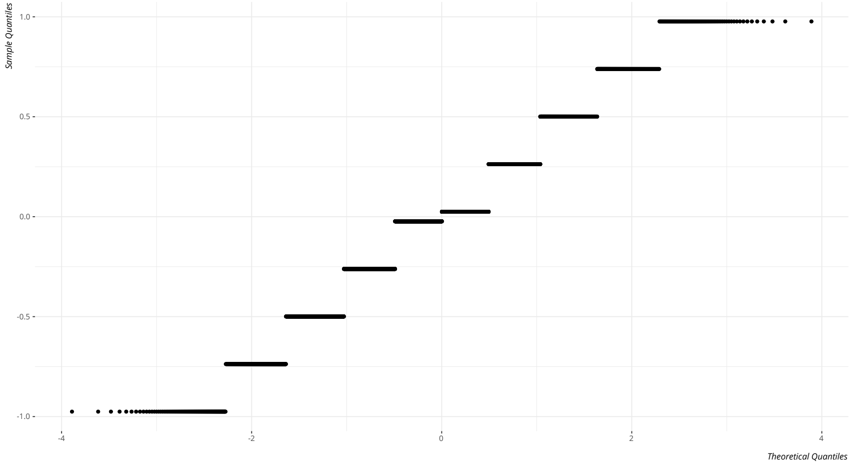
The Fitted-Residual Plot from the OLS Model We Just Ran

No fitted-residual plot from an OLS model should look like this.



The Q-Q Plot from the OLS Model We Just Ran

The Q-Q plot thinks you messed up too, pay careful attention to the middle of the plot as well.



The Right Tool for the Right Job

```r
M2 <- glm(y ~ x, D, family=binomial(link = "logit"))
broom::tidy(M2) %>%
    mutate_if(is.numeric, ~round(., 2)) %>%
    kable(., "markdown")
```

term	estimate	std.error	statistic	p.value
(Intercept)	-2.82	0.06	-48.05	0
x	1.41	0.03	54.24	0

What We Just Did

This was a logistic regression.

- The coefficient tells us the effect of a unit change in `x` on the *natural logged odds of y*.

Let's unpack this piece by piece.

Odds

You typically hear of **odds** in the world of sports betting.

- It's closely linked with probability.

Given some probability *p* of an event occurring, the odds of the event equal:

$$\text{Odds} = \frac{p}{1-p}$$

Ever hear of something like "the odds are 4 to 1 against" an event occurring?

- Translation: for every five trials, we expect 1 occurrence to 4 non-occurrences, on average.
- Odds >1 = more "successes" than "failures."

Probability and Odds in Our Data

```
D %>%
    group_by(x) %>%
    summarize(sum = sum(y),
              length = length(y),
              p = sum/length,
              # q is often substituted as notation for 1 - p
              q = 1 - p,
              odds = p/q) -> sumD
```

Probability and Odds in Our Data

x	sum	length	p	q	odds
0	110	2000	0.06	0.94	0.06
1	399	2000	0.20	0.80	0.25
2	989	2000	0.49	0.51	0.98
3	1609	2000	0.80	0.20	4.12
4	1885	2000	0.94	0.06	16.39

Tell me if you see a pattern beginning to emerge.

Odds Ratio and Percentage Change in Odds

One way of thinking about change in odds is the odds ratio.

- Simply: the odds of y in one category over the odds from the previous category.

A percentage change in odds is also a useful way of seeing a consistent pattern emerge.

- Simply: the difference in odds for a one-unit increase in $\mathtt{x}$ over odds of lower category.

Odds Ratio and Percentage Change in Odds

```
sumD %>%
    mutate(oddsr = odds/lag(odds, 1),
           pcodds = (odds - lag(odds, 1))/lag(odds, 1)*100) -> sumD
```

Percentage Change in Odds

x	sum	length	p	q	odds	oddsr	pcodds
0	110	2000	0.06	0.94	0.06	NA	NA
1	399	2000	0.20	0.80	0.25	4.28	328.20
2	989	2000	0.49	0.51	0.98	3.93	292.52
3	1609	2000	0.80	0.20	4.12	4.21	320.66
4	1885	2000	0.94	0.06	16.39	3.98	298.32

Logit (Natural Logged Odds)

Another more sophisticated/flexible way: logits (i.e. natural logged odds).

- These are natural logarithmic transformations (of base e) of the odds.

Logit (Natural Logged Odds)

```r
sumD %>%
    mutate(logit = log(odds)) %>%
    mutate_if(is.numeric, ~round(., 2)) %>%
    kable(.,"markdown")
```

x	sum	length	p	q	odds	oddsr	pcodds	logit
0	110	2000	0.06	0.94	0.06	NA	NA	-2.84
1	399	2000	0.20	0.80	0.25	4.28	328.20	-1.39
2	989	2000	0.49	0.51	0.98	3.93	292.52	-0.02
3	1609	2000	0.80	0.20	4.12	4.21	320.66	1.41
4	1885	2000	0.94	0.06	16.39	3.98	298.32	2.80

Now do you see it?

Compare M2 with the Previous Table

term	estimate	std.error	statistic	p.value
(Intercept)	-2.82	0.06	-48.05	0
x	1.41	0.03	54.24	0

Properties of Logistic Regression

You can always "backtrack" a logistic regression coefficient.

- Exponentiating a logistic regression coefficient returns an odds ratio. Observe:

```
pull(exp(broom::tidy(M2)[2,2]))
```

```
## [1] 4.0821
```

- You can subtract 1 from the exponentiated coefficient, and multiply it by 100.

```
pull(100*(exp(broom::tidy(M2)[2,2]) - 1))
```

```
## [1] 308.21
```

That's the percentage change in odds.

Properties of Logistic Regression

If you internalize the relationship between probability and odds, you can even return a probability estimate from a logistic regression.

$$\text{Probability} = \frac{\text{Odds}}{1 + \text{Odds}}$$

In R, for when $x = 0$, $pr(y = 1|x = 0)$:.

```
yintercept <- pull(broom::tidy(M2)[1,2])

exp(yintercept)/(1 + exp(yintercept))
```

```
## [1] 0.05629715
```

Properties of Logistic Regression

For larger/more complex models, *resist the urge to do this by hand.*

- But you could if you knew what you were doing.

For example, here's the probability of $y = 1$ for when $x = 2$:

```
yintercept <- pull(broom::tidy(M2)[1,2])
betax <- pull(broom::tidy(M2)[2,2])

exp(yintercept + 2*betax)/(1 + exp(yintercept + 2*betax))
```

```
## [1] 0.4985139
```

Save this train of thought for when we get to the week on making the most of regression.

Properties of Logistic Regression

The logistic function is still monotonic, if not exactly linear.

- Interestingly, logistic function is close to linear when p is between .2 and .8.

Think of the logistic regression function as a natural logged odds of “success.”

- Recall: dummies are a unique case of a categorical variable.

Properties of Logistic Regression

Statistical significance assessments are effectively identical to OLS.

- Caveat: inference is done via *z*-score and not a *t*-statistic.

tl;dr for why: OLS has both a mean and variance to estimate and the variance is independent of the mean.

- In logistic regression, there's really just one parameter p and not two.
- Basically, the variance with binary data is a function of the mean (i.e. $p(1-p)$).

Model Fit for Logistic Regression

Deviance is the estimate of model fit, not R^2.

- Similar to a chi-square analysis.
- i.e. how well does the fitted value ($\hat{y}$) "fit" to the observed value of y.

Bigger the difference (or "deviance"), the poorer the fit of the model.

- This will allow you to do some model comparisons with multiple IVs.

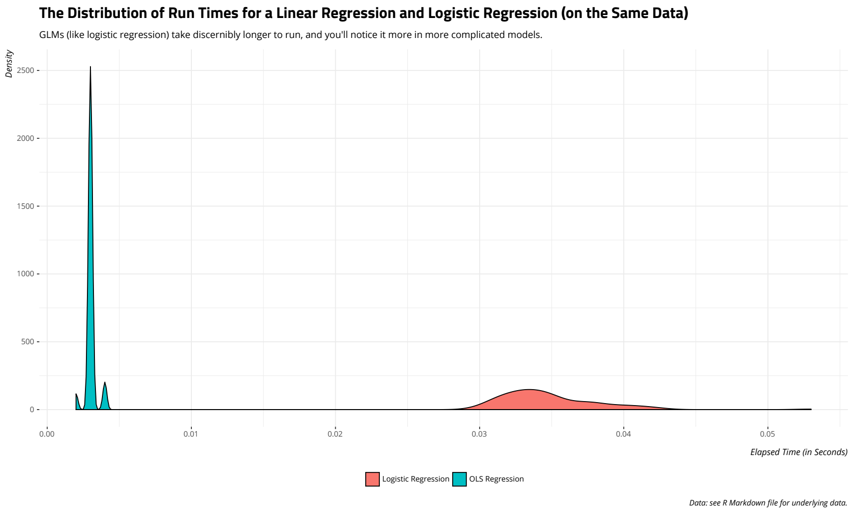
Maximum Likelihood Estimation (MLE)

MLE replaces the OLS principle.

- OLS: draw a line that minimizes the sum of squared residuals.
- MLE: draw a line that results in the smallest possible deviance.

This is done iteratively.

- It's one reason why logistic regression models are slower than linear models.


The Distribution of Run Times for a Linear Regression and Logistic Regression (on the Same Data)
GLMs (like logistic regression) take discernibly longer to run, and you'll notice it more in more complicated models.
Density
2500
2000
1500
1000
500
0
0.00
0.01
0.02
0.03
0.04
0.05
Elapsed Time (in Seconds)
Logistic Regression
OLS Regression
Data: see R Markdown file for underlying data.

Conclusion

If your DV is binary, use a logistic regression and not OLS.

- Statistical signifiance may not change, but that's also not the point.
- Binary DVs violate the assumptions of OLS and produce misleading estimates.
  - *That's the point*.

The process really doesn't change much.

- Inference is done via standard normal distribution, not Student's t-distribution.
- *Coefficients communicate changes in the natural logged odds of* y *for a one-unit change in* x.

This may take some time, but you'll get used to it. I promise.

Table of Contents