PLSC 503: "Multivariate Analysis for Political Research"

Practicum: Models for Binary Responses

April 13, 2017

The topic du jour is interpretation of binary-response model (i.e., logit and probit) estimates.

- Yes, their interpretation is harder than OLS, BUT
- It's not *that* much harder...

We'll talk about a number of different approaches to interpreting these models. But, for now, remember three key points:

1. Nearly all of these approaches require one to be cognizant of "where we are on the curve."
2. When it comes to any kind of interpretation, a picture really is much more valuable than text or numbers.
3. With very rare exceptions, it is never a good idea to present quantities of interest without their associated measures of uncertainty.

A Running Example: House Voting on NAFTA

To motivate the discussion, we'll use a running example: The U.S. House of Representatives vote on the North American Free Trade Agreement (NAFTA). In 1993, the House voted to approve ratification of NAFTA by a margin of 234-200. Our example data thus contain 435 observations and five variables:

1. `vote` – Whether (=1) or not (=0) the House member in question voted in favor of NAFTA.
2. `democrat` – Whether the House member in question is a Democrat (=1) or a Republican (=0).
3. `pcthispc` – The percentage of the House member's district who are of Latino/hispanic origin.
4. `cope93` – The 1993 AFL-CIO (COPE) voting score of the member in question; this variable ranges from 0 to 100, with higher scores indicating more pro-labor positions.
5. `DemXCOPE` – The multiplicative interaction of `democrat` and `cope93`.

Our expectations are that:

- Higher COPE scores will correspond to lower probabilities of voting for NAFTA,

- Members from districts with higher numbers of Latinos will have higher probabilities of voting for NAFTA, but
- The effect of the former will be moderated by political party. In particular, the (negative) effect of COPE scores on pro-NAFTA voting will be greater for Democrats than for Republicans.

The relevant model, then, looks like:

$$\begin{aligned}\Pr(\texttt{vote}_i = 1) &= f[\beta_0 + \beta_1(\texttt{democrat}_i) + \beta_2(\texttt{pcthispc}_i) + \\ &\quad \beta_3(\texttt{cope93}_i) + \beta_4(\texttt{democrat}_i \times \texttt{cope93}_i) + u_i] \end{aligned} \tag{1}$$

The data look like this:

```
      vote            democrat          pcthispc         cope93           DemXCOPE
 Min.   :0.0000   Min.   :0.0000   Min.   : 0.0   Min.   :  0.00   Min.   :  0.00
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 1.0   1st Qu.: 17.00   1st Qu.:  0.00
 Median :1.0000   Median :1.0000   Median : 3.0   Median : 81.00   Median : 75.00
 Mean   :0.5392   Mean   :0.5853   Mean   : 8.8   Mean   : 60.18   Mean   : 51.65
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:10.0   3rd Qu.:100.00   3rd Qu.:100.00
 Max.   :1.0000   Max.   :1.0000   Max.   :83.0   Max.   :100.00   Max.   :100.00
```

and we can estimate this model using (e.g.) the `glm` command in R :

```
> fit<-glm(vote~democrat+pcthispc+cope93+DemXCOPE,family=binomial)
> summary(fit)

Call:
glm(formula = vote ~ democrat + pcthispc + cope93 + DemXCOPE,
    family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.791640   0.275438   6.505 7.79e-11 ***
democrat     6.865556   1.547295   4.437 9.12e-06 ***
pcthispc     0.020911   0.007941   2.633  0.00846 **
cope93      -0.036501   0.007598  -4.804 1.55e-06 ***
DemXCOPE    -0.067054   0.018203  -3.684  0.00023 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 598.99  on 433  degrees of freedom
Residual deviance: 436.83  on 429  degrees of freedom
AIC: 446.83

Number of Fisher Scoring iterations: 5
```

For the rest of the class, we'll talk about a host of means for interpreting models, using these data as a running example.

"Signs-N-Significance"

One alternative for interpretation is what I call "signs-n-significance": talk about the sign(s) of the estimated coefficient(s), and whether (and to what extent) that coefficient is statistically differentiable from zero.

- For all of the models we've discussed, this approach is no different than for OLS:
  - A positive estimate for $\hat{\beta}_X$ mean that increases in X correspond to increases in $\Pr(Y = 1)$.
  - Likewise, a negative estimate for $\hat{\beta}_X$ mean that increases in X correspond to decreases in $\Pr(Y = 1)$.
- Similarly, the ratio of $\hat{\beta}$ to its standard error is a z-score that can be used for hypothesis testing, etc.

So, in the example, we might note that (for Republicans), the estimate of the effect of `cope93` is negative (as expected), and that it is "statistically significant" (because its z-score is -4.8).

Note also that, because we have an interaction term in the model, the "direct effects" have a conditional interpretation. So, for example (referencing Equation 1), the estimated coefficient for the `cope93` variable when `democrat = 1` is:

$$\hat{\beta}_{\texttt{cope93|democrat=1}} = \hat{\beta}_3 + \hat{\beta}_4$$

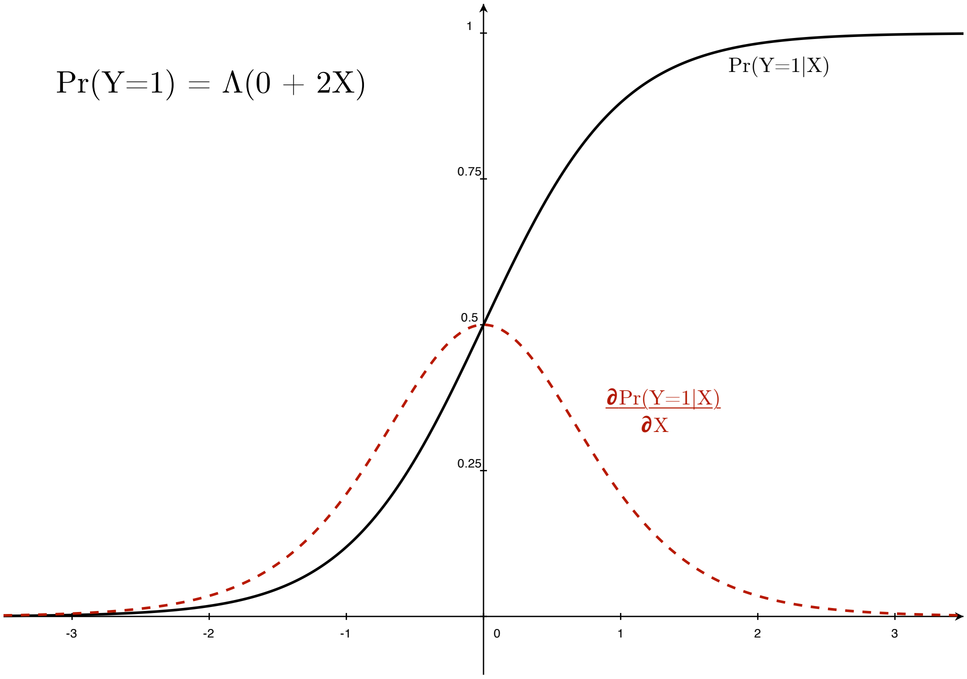
As with linear regression, we can obtain these "conditional" coefficient estimates – and their standard errors, z-scores, and confidence intervals – using R :

```
> fit$coeff[4]+fit$coeff[5]
    cope93
-0.1035551

> (fit$coeff[4]+fit$coeff[5]) / (sqrt(vcov(fit)[4,4] + (1)^2*vcov(fit)[5,5] +
  2*1*vcov(fit)[4,5]))
```

```
   cope93
-6.245699
```

Thus, we can say that the effect of pro-union ideology – which was negative and significant for Republicans – is also negative, also significant, and roughly three times larger for Democrats.

Figure 1: $\Pr(Y=1)$ and $\frac{\partial \Pr(Y=1)}{\partial X}$ versus X for $Y = \Lambda(0 + 2X)$



Predicted Probabilities

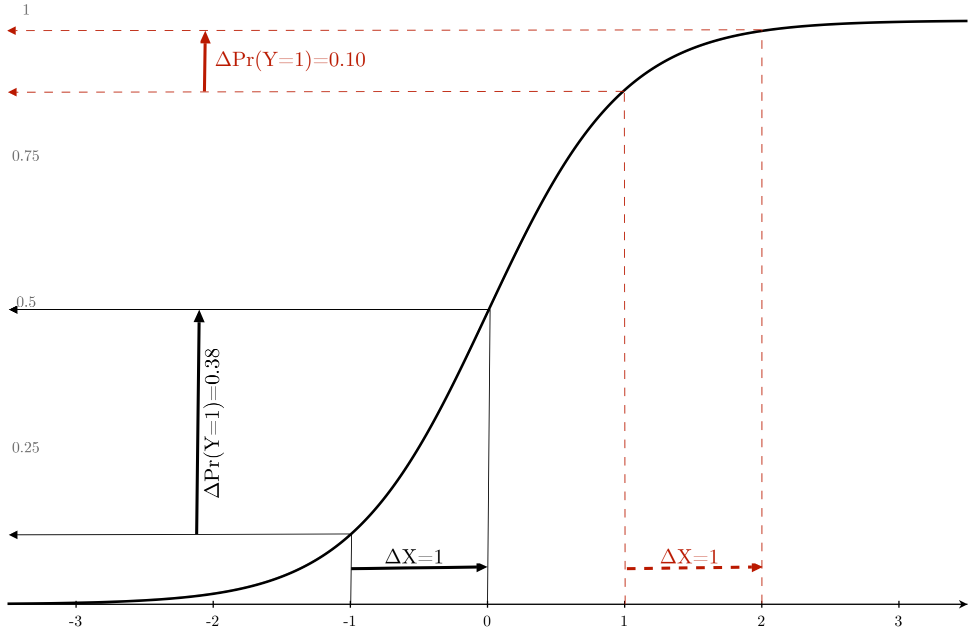
"Signs-n-Significance" is – to put it mildly – a rotten way to interpret this or any statistical model. What we really care about is, in most cases, the effect of changes in $\mathbf{X}$ on $\Pr(Y = 1)$ – that is, on the probability of the actual event of interest. To get at this is a bit more involved than in the OLS case.

In all the binary-response models we've discussed, the effect of covariates is *linear in the latent variable* (that is, Y^*), but not in Y. *The real net effect of a change in X depends critically on the values of the other Xs and parameter estimates, and on the constant;* this is because the model is *nonlinear*. We can see this by noting that – unlike in a linear regression model – the first derivative of a logit/probit function depends on the value(s) of X and $\hat{\beta}$. So, for example, in the case of a binary logit:

$$\frac{\partial \Pr(\hat{Y}_i = 1)}{\partial X_k} \equiv \lambda(X_k) = \frac{\exp(X_i\hat{\beta})}{[1 + \exp(X_i\hat{\beta})]^2}\hat{\beta}_k \tag{2}$$

This non-constant first derivative is illustrated in Figure 1. As a practical matter, this means that if you're interested in the effect of a one-unit change in X on $\Pr(Y_i = 1)$, how much change there is depends critically on "where you are on the curve."

Figure 2: Changes in $\Pr(Y = 1)$ for One-Unit Changes in X, for $Y = \Lambda(0 + 2X)$



To illustrate this, consider first a model with a single continuous covariate X:

$$\Pr(Y_i = 1) = \Lambda(\beta_0 + \beta_1 X_i) \tag{3}$$

where we have an estimated $\hat{\beta}_1 = 2.0$ and no intercept ($\hat{\beta}_0 = 0$). The change in $\Pr(Y = 1)$ associated with a one-unit change in X varies depending on the "baseline" value of X, as well as on any other covariates' values in the model. So, for example, if we change $X = -1$ to $X = 0$, then the associated change in the predicted probability is:

$$\begin{aligned}
\frac{\exp(2\times 0)}{1+\exp(2\times 0)} - \frac{\exp(2\times -1)}{1+\exp(2\times -1)} &= \frac{\exp(0)}{1+\exp(0)} - \frac{\exp(-2)}{1+\exp(-2)} \\
&= \frac{1}{2} - \frac{0.14}{1.14} \\
&= 0.50 - 0.12 \\
&= \mathbf{0.38}
\end{aligned}$$

On the other hand, if we are going from $X = 1$ to $X = 2$, we get:

$$\begin{aligned}
\frac{\exp(2\times 2)}{1+\exp(2\times 2)} - \frac{\exp(2\times 1)}{1+\exp(2\times 1)} &= \frac{\exp(4)}{1+\exp(4)} - \frac{\exp(2)}{1+\exp(2)} \\
&= \frac{7.39}{8.39} - \frac{1}{2} \\
&= 0.98 - 0.88 \\
&= \mathbf{0.10}
\end{aligned}$$

These changes are illustrated graphically in Figure 2.

More generally, the change in $\Pr(Y = 1)$ associated with a (possibly multivariate) change in the values of the covariate vector $\mathbf{X}$ from $\mathbf{X}_A$ to $\mathbf{X}_B$ is:

$$\Delta\Pr(Y=1)_{\mathbf{X}_A\to\mathbf{X}_B} = \frac{\exp(\mathbf{X}_B\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{X}_B\hat{\boldsymbol{\beta}})} - \frac{\exp(\mathbf{X}_A\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{X}_A\hat{\boldsymbol{\beta}})} \tag{4}$$

As a practical matter, all this means that you need to be careful of the values of the other variables in the model when assessing one variable's impact.

Getting Predicted Probabilities

All predicted probabilities – whether in-sample or out-of-sample – take on the same general form:

$$\begin{aligned}
\Pr(Y_i = 1) &= \frac{\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}})} \text{ for logit,} \\
&= \Phi(\mathbf{X}_i\hat{\boldsymbol{\beta}}) \text{ for probit.}
\end{aligned}$$

R makes it very easy to obtain these predictions after estimating your regression. After running either model, just type:

```
> preds<-fit$fitted.values
```

...and R will create an object called `preds` that contains the predicted probability of $Y_i = 1$ for each of the observations in whatever data are used for the analysis. Also useful is the fact that we can generate the predicted *index value* $\mathbf{X}_i\hat{\boldsymbol{\beta}}$ for each observation in the data, using:

```
> hats<-predict(fit,se.fit=TRUE)
> hats
$fit
          1           2           3           4 ...
 9.01267619  7.25223902  6.11013844  5.57444635 ...
 ...
 $se.fit
        1         2         3         4 ...
1.5331506 1.2531475 1.1106989 0.9894208 ...
```

As you can see, using the `se.fit=TRUE` also generates the *standard error of the linear prediction* – that is, the standard error of $\mathbf{X}_i\hat{\boldsymbol{\beta}}$. Note that the variance of the predicted value $\Lambda(\mathbf{X}_i\hat{\boldsymbol{\beta}})$ is

$$\begin{aligned} \text{Var}[\text{Pr}\widehat{(Y_i = 1)})] &= \left[\frac{\partial F(\mathbf{X}_i\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}}\right]' \hat{\mathbf{V}} \left[\frac{\partial F(\mathbf{X}_i\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}}\right] \\ &= [f(\mathbf{X}_i\hat{\boldsymbol{\beta}})]^2\mathbf{X}_i'\hat{\mathbf{V}}\mathbf{X}_i \end{aligned}$$

which means the standard error of that prediction is just:

$$\text{s.e.}[\text{Pr}\widehat{(Y_i = 1)})] = \sqrt{[f(\mathbf{X}_i\hat{\boldsymbol{\beta}})]^2\mathbf{X}_i'\hat{\mathbf{V}}\mathbf{X}_i}$$

As we'll see, these standard errors are important when we go to use these predictions.[1]

What Do We Do With Predictions?

First off, note that there are two types of predictions that we typically care about:

- *In-sample predictions* are simply the predicted probabilities of $Y_i = 1$ for the observations in the data on which the model was estimated.
- *Out-of-sample predictions* are predictions for cases that don't necessarily exist in the data, but which might be of interest (e.g., hypothetical cases).

[1] As a matter of fact, there are lots of other things that `predict` will calculate for you after estimation of probit or logit models; these include various kinds of residuals, influence statistics (to check for outliers), score values, etc. We won't go into those here; check `?predict` in R if you're curious...

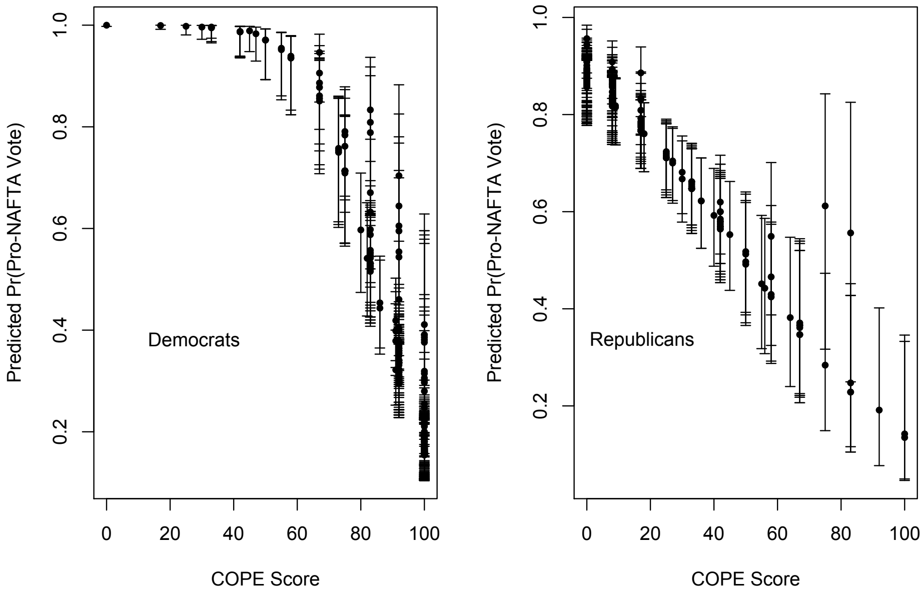
In-Sample Predictions

As we note above, R can generate in-sample predictions and their associated measures of uncertainty using `predict()`. For our NAFTA voting data, for example, we might use:

```
> XBUB<-hats$fit + (1.96*hats$se.fit)
> XBLB<-hats$fit - (1.96*hats$se.fit)
> plotdata<-cbind(as.data.frame(hats),XBUB,XBLB)
> plotdata<-data.frame(lapply(plotdata,binomial(link="logit")$linkinv))
> par(mfrow=c(1,2))
> library(plotrix)
> plotCI(cope93[democrat==1],plotdata$fit[democrat==1],ui=plotdata$XBUB[democrat==1],
  li=plotdata$XBLB[democrat==1],pch=20,xlab="COPE Score",ylab="Predicted
  Pr(Pro-NAFTA Vote)")
> plotCI(cope93[democrat==0],plotdata$fit[democrat==0],ui=plotdata$XBUB[democrat==0],
  li=plotdata$XBLB[democrat==0],pch=20,xlab="COPE Score",ylab="Predicted
  Pr(Pro-NAFTA Vote)")
```

which generates Figure 3:

Figure 3: Smoothed In-Sample $\widehat{\Pr(Y=1)}$s and 95% Confidence Intervals



These plots are based on the actual data, and so are very "noisy;" notice that I use a median spline smoother to make the plots smoother (and prettier). The plot in Figure 3 is informative; for example, it clearly reflects the finding that the effect of changes in `cope93` are greater for Democrats than for Republicans.

Out-Of-Sample Predictions

We also use the `-predict-` command to do out-of-sample predictions. The key to dealing with out-of-sample predictions is in selecting the variables of most interest to us, and then holding the other variables in the model constant at some particular level. To do this, we usually choose the mean for continuous variables, and the median or mode for others (since the mean of a dichotomous variable is a nonsensical value). We then use these to calculate an *index value*: the value of the combination of the covariate mean/medians and the estimated coefficients (i.e., $\sum_{k=1}^{K} \overline{X}_k \hat{\beta}_k$); to get the estimated mean probability of a positive outcome, we take the logit transform of this.

Presenting out-of-sample predictions usually involves one of two methods: either we report the predictions in *tabular* format, or we *plot* the predicted probabilities. 99 percent of the time, the latter is better than the former, but we'll discuss both here.

Tables of Predictions

The method for generating tables of predicted probabilities is pretty straightforward:

1. Calculate the "index value" for a "mean" (typical) observation.
2. Choose some amount by which you want to change the value of the independent variable in question.
    - Standard deviations are good – they make results across covariates a little more comparable. But,
    - They may also be some substantively interesting value(s).
3. Calculate the "index value" for two (or more) alternative observations – say, one σ above and below the mean.
4. From these, calculate the associated predicted probabilities, and report them.

All of this can be done (e.g.) in a spreadsheet, to make your life easier.

Graphs of Predicted Probabilities

Generally, plots of predicted probabilities are a (much) better option than tables. First, pick out a variable of interest: Let's assume here that we are interested in plotting graphically the different effects of changes in COPE scores on Democrats' and Republicans' votes for NAFTA. To do this, we start with a "dummy" (simulated) dataset that contains all the variables used in the original analysis:

```
> sim.data<-data.frame(pctthispc=mean(nafta$pcthispc),democrat=rep(0:1,101),
  cope93=seq(from=0,to=100,length.out=101))
> sim.data$DemXCOPE<-sim.data$democrat*sim.data$cope93
```

These simulated data contain all the (independent) variables in the original analysis. Note that we have set the pcthispc variable to its mean value. We also have 101 observations in which democrat=0 and cope93 ranges from 0 to 100, and another 101 observations in which democrat=1 and cope93 ranges from 0 to 100.

Next, we return to the original model, and use predict on our simulated data to generate predicted probabilities and their standard errors, in a manner analogous to what we did in-sample:

```
> OutHats<-predict(NAFTA.GLM.logit,se.fit=TRUE,newdata=sim.data)
> OutHatsUB<-OutHats$fit+(1.96*OutHats$se.fit)
> OutHatsLB<-OutHats$fit-(1.96*OutHats$se.fit)
> OutHats<-cbind(as.data.frame(OutHats),OutHatsUB,OutHatsLB)
> OutHats<-data.frame(lapply(OutHats,binomial(link="logit")$linkinv))
```

We now have the predicted probabilities and their 95 percent confidence intervals for the range of cope93 values, for both Democrats and Republicans. We can use these data to create plots, like the one in Figure 4:

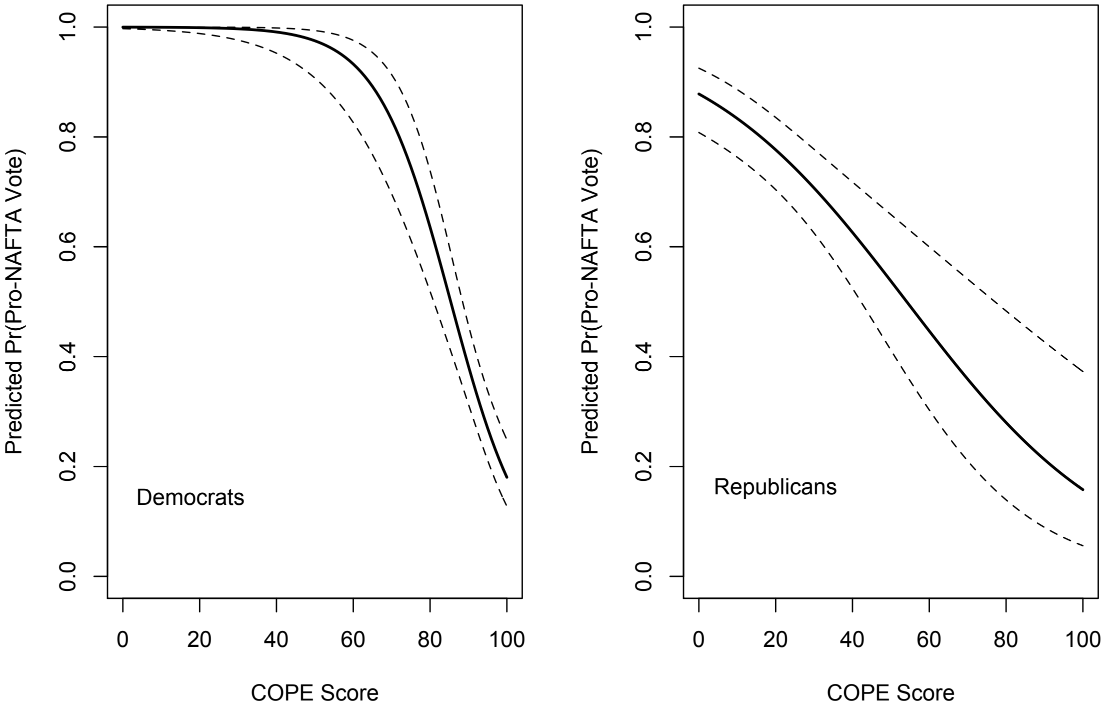
```
> par(mfrow=c(1,2))
> both<-cbind(sim.data,OutHats)
> both<-both[order(both$cope93,both$democrat),]

> plot(both$cope93[democrat==1],both$fit[democrat==1],t="l",lwd=2,ylim=c(0,1),
  xlab="COPE Score",ylab="Predicted Pr(Pro-NAFTA Vote)")
> lines(both$cope93[democrat==1],both$OutHatsUB[democrat==1],lty=2)
> lines(both$cope93[democrat==1],both$OutHatsLB[democrat==1],lty=2)
> text(locator(1),label="Democrats")

> plot(both$cope93[democrat==0],both$fit[democrat==0],t="l",lwd=2,ylim=c(0,1),
  xlab="COPE Score",ylab="Predicted Pr(Pro-NAFTA Vote)")
> lines(both$cope93[democrat==0],both$OutHatsUB[democrat==0],lty=2)
> lines(both$cope93[democrat==0],both$OutHatsLB[democrat==0],lty=2)
> text(locator(1),label="Republicans")
```

Figure 4 tells us that, with all other variables at their mean values, the predicted probability of a pro-NAFTA vote decreases significantly as a member's COPE score increases. However, the magnitude of that decrease is significantly larger for Democrats than for Republicans. One interpretation of this, then, is that Democrats are (as one might expect) more responsive to district-level union concerns than are Republicans.

The confidence intervals in Figure 4 are also useful. They tell us that, over most of the range of COPE scores, Democrats and Republicans are quite different in their probability of voting for NAFTA. At high levels of COPE scores, however, they converge. Thus, pro-union Democrats and Republicans tended to vote alike (that is, against NAFTA), while those less favorable toward unions behaved differently.

There are other things one can do with predicted probabilities, including 3-D plots (which are very useful for interactions of two continuous variables). But that's probably enough for now; let's move on.

Figure 4: Predicted Probabilities of a Pro-NAFTA Vote, by Party Identification and COPE Score



Note: Solid line is predicted probabilities for Republicans; dashed line for Democrats. Regions indicate 95% pointwise confidence intervals for the predictions.

Odds Ratios and the Logit Model

Odds ratios are an easy way of substantively interpreting a logit model. Consider the **"odds"** of $Y = 1$ for a given observation with some values of $\mathbf{X}$:

$$\Omega(\mathbf{X}) = \frac{\Pr(Y=1|\mathbf{X})}{\Pr(Y=0|\mathbf{X})} = \frac{\Pr(Y=1|\mathbf{X})}{1-\Pr(Y=1|\mathbf{X})} = \frac{\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}}{1-\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}} \tag{5}$$

So, if $\Pr(Y = 1) = 0.50$, then the odds are 1 to 1; for $\Pr(Y = 1) = 0.75$, the odds are 3 to 1, etc. Note that the odds range from zero to infinity (but will never be negative).

If Ω is the odds, then $\ln \Omega$ is the "log-odds" (which is also known as the *logit*), which ranges from negative infinity to infinity. This means that:

$$\ln \Omega(\mathbf{X}) = \ln \left[\frac{\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}}{1-\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}} \right] = \mathbf{X}\boldsymbol{\beta} \tag{6}$$

That is – and as we noted last class – in the logit model, the log-odds are linear in $\mathbf{X}$. This means that if we consider the effect of a change in $\mathbf{X}$ on the log-odds of Y, we get:

$$\frac{\partial \ln \Omega}{\partial \mathbf{X}} = \boldsymbol{\beta} \tag{7}$$

In other words, the estimate $\hat{\beta}_k$ from our logit equation tells us the change in the log-odds which accompanies a one-unit change in X_k.

But normal people don't think in terms of log-odds,[2] they think in terms of odds. (7) means that the change in the odds of $Y = 1$ associated with a one-unit change in X_k is:

$$\frac{\Omega(X_k+1)}{\Omega(X_k)} = \exp(\hat{\beta}_k) \tag{8}$$

More generally,

$$\frac{\Omega(X_k+\delta)}{\Omega(X_k)} = \exp(\hat{\beta}_k\delta) \tag{9}$$

As a practical matter, Equation (8) means that we can interpret the exponentiated coefficients of a logit model as the change in the odds of $\Pr(Y = 1)$ associated with a one-unit change in X_k. This translates easily to a percentage change in the odds as well:

$$\text{Percentage Change} = 100[\exp(\hat{\beta}_k\delta) - 1] \tag{10}$$

Thus, for example:

- For a logit estimate of $\hat{\beta} = 2.3$, a unit change in X...
    - ...corresponds to an increase in the log-odds of $Y = 1$ of 2.3, or

[2] If I took you to the track, and told you the log-odds of a horse coming in win, place or show were -1.95, would you bet on it? (That's 7 to one, for you race fans...).

  - ...a change in the odds that $Y = 1$ of $\exp(2.3) = 9.974$, or
  - ...a percentage change in the odds that $Y = 1$ of $100[\exp(\hat{\beta}) - 1] = 897$ percent.

- For a logit estimate of $\hat{\beta} = -0.22$, an 11–unit change in X...
  - ...corresponds to a $-0.22 \times 11 = -2.42$ decrease in the log-odds of $Y = 1$, or
  - ...a change in the odds that $Y = 1$ of $\exp(-0.22 \times 11) = \exp(-2.42) = 0.089$, or
  - ...a percentage change in the odds that $Y = 1$ of $100[\exp(-0.22 \times 11) - 1] = 100(0.089 - 1) = -91.1$ percent.

Odds ratios are thus an easy, intuitive way to interpret logit coefficients.[3] Doing so using `glm` requires a bit of tweaking, via this function:

```
> lreg.or <- function(model)
+           {
+           coeffs <- coef(summary(NAFTA.GLM.logit))
+           lci <- exp(coeffs[ ,1] - 1.96 * coeffs[ ,2])
+           or <- exp(coeffs[ ,1])
+           uci <- exp(coeffs[ ,1] + 1.96 * coeffs[ ,2])
+           lreg.or <- cbind(lci, or, uci)
+           lreg.or
+           }
```

We can then show the odds ratios by aiming that function at our `glm` object:

```
> lreg.or(NAFTA.GLM.fit)
                lci       or       uci
(Intercept)  3.4966   5.9993 1.029e+01
democrat    46.1944 958.6783 1.990e+04
pcthispc     1.0054   1.0211 1.037e+00
cope93       0.9499   0.9642 9.786e-01
DemXCOPE     0.9024   0.9351 9.691e-01
```

Finally, in using odds ratios, remember that:

- Percentage decreases in odds are bounded at 100 (naturally), but have no upper bound. That means that...

[3] For Stata users out there, the software will report odds ratios rather than $\hat{\boldsymbol{\beta}}$s – along with their standard errors, z-scores, and confidence intervals – automatically; all you have to do is ask.

- ...if $\exp(\hat{\beta}_k\delta) < 1$, the we would say that the odds of $Y = 1$ are "only $100[\exp(\hat{\beta}_k\delta) - 1]$ percent of those for cases with $X = X_0 + \delta$, versus those with X_0."
- If $\exp(\hat{\beta}_k\delta) > 1$, the we would say that the odds of $Y = 1$ are "$100[\exp(\hat{\beta}_k\delta) - 1]$ percent of those for cases with $X = X_0 + \delta$, versus those with X_0."