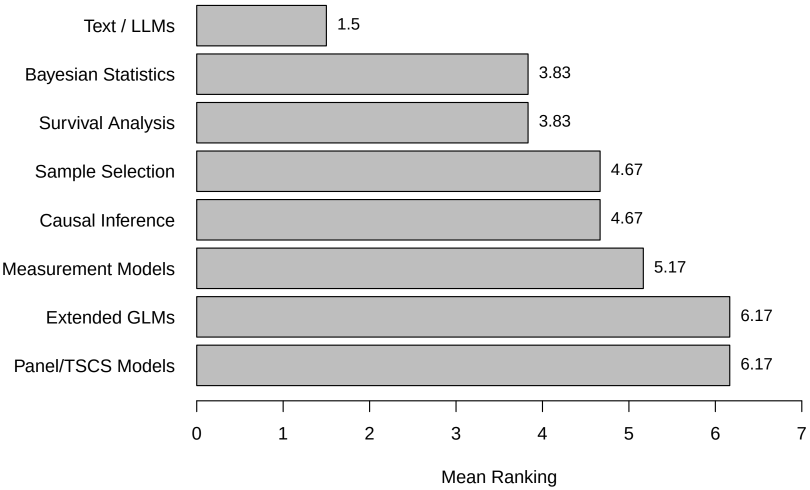
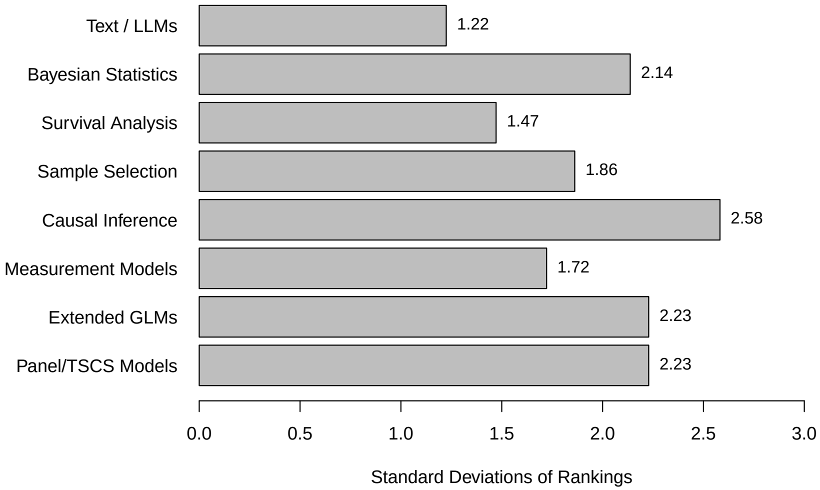
GSERM 2024
Regression for Publishing

June 20, 2024

Participants' Choice: Rankings



Participants' Choice: Standard Deviations



Catching Up...

Digression II: The Random Utility Model

$$Y \in \{SQ, A\}$$

$$\begin{aligned} Y_i &= A \quad \text{if} \quad \mathrm{E}[\mathrm{U}_i(A)] \geq \mathrm{E}[\mathrm{U}_i(SQ)] \\ &= SQ \quad \text{if} \quad \mathrm{E}[\mathrm{U}_i(A)] < \mathrm{E}[\mathrm{U}_i(SQ)] \end{aligned}$$

$$\mathrm{E}[\mathrm{U}_i(A)] = \mathbf{X}_{iA}\boldsymbol{\beta} + u_{iA}$$

So:

$$\begin{aligned} \Pr(Y = A) &= \Pr\{\mathrm{E}[\mathrm{U}_i(A)] \geq \mathrm{E}[\mathrm{U}_i(SQ)]\} \\ &= \Pr\{(\mathbf{X}_{iA}\boldsymbol{\beta} + u_{iA}) \geq \mathrm{E}[\mathrm{U}_i(SQ)]\} \end{aligned}$$

Digression II: The Random Utility Model

Normalize:

$$\mathrm{E}[\mathrm{U}_i(SQ)] = 0$$

Then:

$$\begin{aligned}
\Pr(Y = A) &= \Pr\{(\mathbf{X}_{iA}\boldsymbol{\beta} + u_{iA}) \geq 0\} \\
&= \Pr\{u_{iA} \geq -\mathbf{X}_{iA}\boldsymbol{\beta}\} \\
&= F(\mathbf{X}_{iA}\boldsymbol{\beta})
\end{aligned}$$

Another Model: Complementary Log-Log

Uses:

$$\Pr(Y_i = 1) = 1 - \exp[-\exp(\mathbf{X}_i\beta)]$$

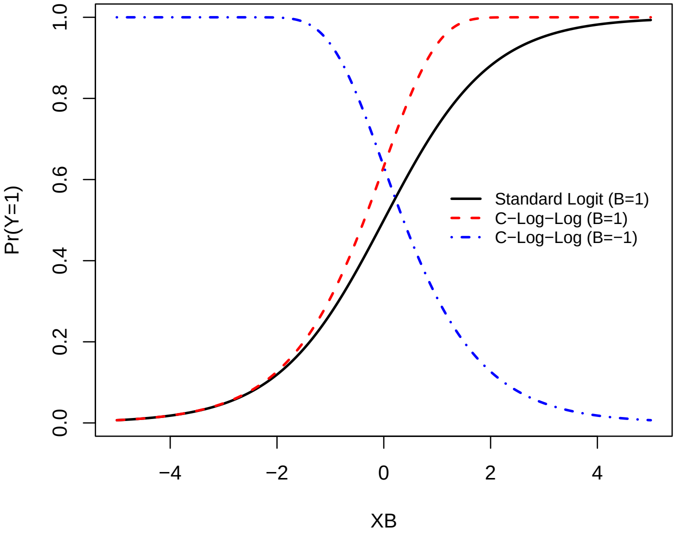
or

$$\ln\{-\ln[1 - \Pr(Y_i = 1)]\} = \mathbf{X}_i\beta$$

Likelihood is:

$$\begin{aligned}\ln L &= \sum_{i=1}^{N} Y_i \ln\{1 - \exp[-\exp(\mathbf{X}_i\beta)]\} + \\ &\quad (1 - Y_i)\ln\{1 - \{1 - \exp[-\exp(\mathbf{X}_i\beta)]\}\}\end{aligned}$$

Logit and C-log-log CDFs



All require that:

- "Threshold" $= Y^* > 0$
- $\mathrm{E}(u_i|\mathbf{X}, \boldsymbol{\beta}) = 0$
- $\mathrm{Var}(u_i) = \frac{\pi^2}{3}$ or 1.0.

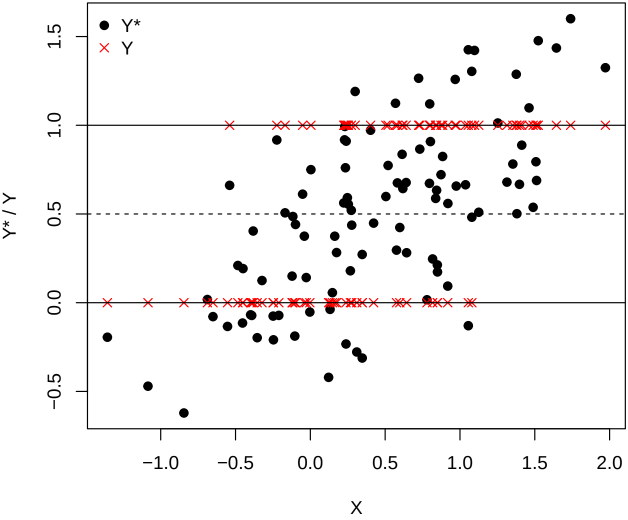
In general:

- The Universe: Logit $>$ Probit
- The (Social Science) Universe: Meh...
- $\hat{\beta}_{\text{Logit}} \approx 1.8 \times \hat{\beta}_{\text{Probit}}$
- Four reasons to prefer / use logit

A Toy Example

```
> set.seed(7222009)
> ystar<-rnorm(100,0.5,0.5)
> y<-ifelse(ystar>0.5,1,0)
> x<-ystar+(0.5*rnorm(100))
> data<-data.frame(ystar,y,x)

> head(data)
     ystar y       x
1  0.17977 0  0.2677
2  0.79428 1  1.5079
3  0.82408 1  0.8842
4  0.24658 0  0.8172
5  0.50966 1  1.1255
6 -0.07852 0 -0.6506
```

A Toy Example



```
> mylogit<-glm(y~x,family=binomial(link="logit"), data=data)
> myprobit<-glm(y~x,family=binomial(link="probit"), data=data)
> mycloglog<-glm(y~x,family=binomial(link="cloglog"), data=data)
```

Logit, Probit, and C-Log-Log Models (Simulated Data)

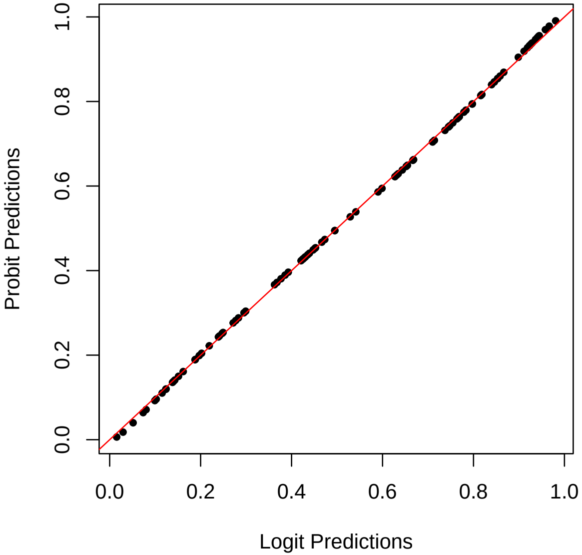
	Logit	*Probit*	*C-Log-Log*
X	2.428^{***}	1.458^{***}	1.613^{***}
	(0.500)	(0.272)	(0.309)
Constant	-0.861^{***}	-0.519^{***}	-1.048^{***}
	(0.318)	(0.183)	(0.250)
Observations	100	100	100
Log Likelihood	-49.690	-49.490	-49.522
Akaike Inf. Crit.	103.380	102.979	103.044
Note:	*p$<$0.1; **p$<$0.05; ***p$<$0.01		

Comparing Models (continued)

Note:

- zs, Ps, $\ln L$s, AICs nearly identical
- Residuals, too
- $\hat{\beta}_{\text{Logit}}$ is $\frac{2.428}{1.458} = 1.54 \times \hat{\beta}_{\text{Probit}}$

Toy Example: Predicted Probabilities



Note: C-Log-Log Isn't "Reversible"

Suppose we generate a new dependent variable:

$$Y_{i\text{New}} = 1 - Y_i$$

What happens to our estimates?

	$\hat{\beta}_0$			$\hat{\beta}_1$		
	Y		Y_{New}	Y		Y_{New}
Probit	-0.52	$\leftrightarrow$	0.52	1.46	$\leftrightarrow$	-1.46
Logit	-0.86	$\leftrightarrow$	0.86	2.43	$\leftrightarrow$	-2.43
C-Log-Log	-1.05	$\leftrightarrow$	0.11	1.61	$\leftrightarrow$	-1.66

Practical Binary Response Models

Running Example: House Vote on NAFTA (1993)

Response / Outcome

- `vote` – Whether (=1) or not (=0) the U.S. House member in question voted in favor of NAFTA.

Predictors

- `PropHisp` – The proportion of the House member's district who are of Latino/hispanic origin.
- `Democrat` – Whether the House member in question is a Democrat (=1) or a Republican (=0).
- `COPE` – The 1993 AFL-CIO (COPE) voting score of the member in question; the original variable ranges from 0 to 100, with higher scores indicating more pro-labor positions. Rescaled to range from 0 to 1.
- `DemXCOPE` – The multiplicative interaction of `Democrat` and `COPE`.

The model:

$$\Pr(\texttt{vote}_i = 1) = f[\beta_0 + \beta_1(\texttt{PropHisp}_i) + \beta_2(\texttt{Democrat}_i) + \beta_3(\texttt{COPE}_i) + \beta_4(\texttt{Democrat}_i \times \texttt{COPE}_i) + u_i]$$

The data:

```
> describe(NAFTA,fast=TRUE,skew=TRUE)

         vars   n mean   sd median min  max range  skew kurtosis   se
Vote        1 434 0.54 0.50   1.00   0 1.00  1.00 -0.16    -1.98 0.02
PropHisp    2 434 0.09 0.14   0.03   0 0.83  0.83  2.76     7.74 0.01
Democrat    3 434 0.59 0.49   1.00   0 1.00  1.00 -0.34    -1.89 0.02
COPE        4 434 0.60 0.39   0.81   0 1.00  1.00 -0.42    -1.52 0.02
DemXCOPE    5 434 0.52 0.46   0.75   0 1.00  1.00 -0.17    -1.87 0.02
```

Probit Estimates

```
> NAFTA.probit<-glm(Vote~PropHisp+Democrat+COPE+DemXCOPE,
                    NAFTA,family=binomial(link="probit"))
> summary(NAFTA.probit)

Call:
glm(formula = Vote ~ PropHisp + Democrat + COPE + DemXCOPE,
    family = binomial(link = "probit"), data = NAFTA)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.173  -0.677   0.362   0.764   1.817

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.078      0.153    7.03  2.1e-12 ***
PropHisp       1.279      0.467    2.74   0.0062 **
Democrat       3.034      0.739    4.11  4.0e-05 ***
COPE          -2.201      0.440   -5.00  5.8e-07 ***
DemXCOPE      -2.888      0.903   -3.20   0.0014 **
---
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: 441.06  on 429  degrees of freedom
AIC: 451.1

Number of Fisher Scoring iterations: 8
```

Logit Estimates

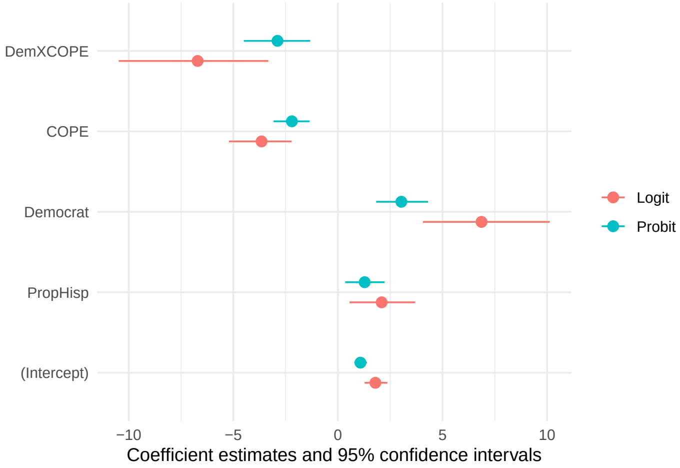
```
> NAFTA.fit<-glm(Vote~PropHisp+Democrat+COPE+DemXCOPE,
                       NAFTA,family=binomial)
> summary(NAFTA.fit)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.264  -0.650   0.310   0.728   1.818

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.792      0.275    6.50  7.8e-11 ***
PropHisp       2.091      0.794    2.63  0.00846 **
Democrat       6.866      1.547    4.44  9.1e-06 ***
COPE          -3.650      0.760   -4.80  1.6e-06 ***
DemXCOPE      -6.705      1.820   -3.68  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: 446.8

Number of Fisher Scoring iterations: 5

> # Equivalent to:
>
> fit<-glm(Vote~PropHisp+Democrat*COPE,NAFTA,family=binomial)
```

Models (table via `modelsummary`)

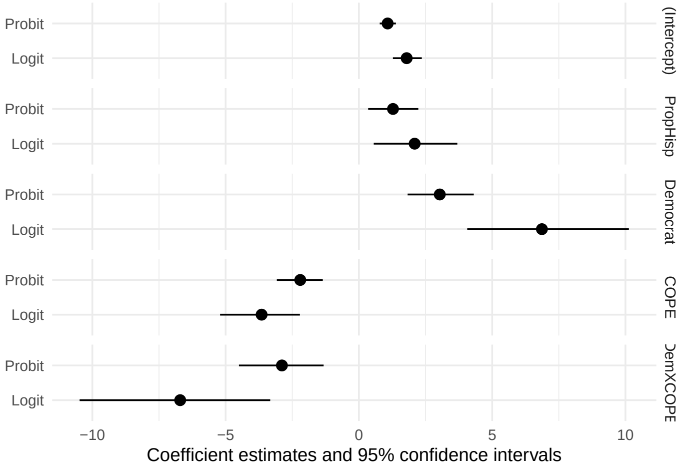
Table 1: Logits and Probits

	Logit	Probit
(Intercept)	1.792	1.078
	(0.275)	(0.153)
Proportion Hispanic	2.091	1.279
	(0.794)	(0.467)
Democrat	6.866	3.034
	(1.547)	(0.739)
COPE Score	-3.650	-2.201
	(0.760)	(0.440)
Democrat x COPE Score	-6.705	-2.888
	(1.820)	(0.903)
Num.Obs.	434	434
AIC	446.8	451.1
BIC	467.2	471.4
Log.Lik.	-218.414	-220.532
F	26.622	30.723
RMSE	0.40	0.41

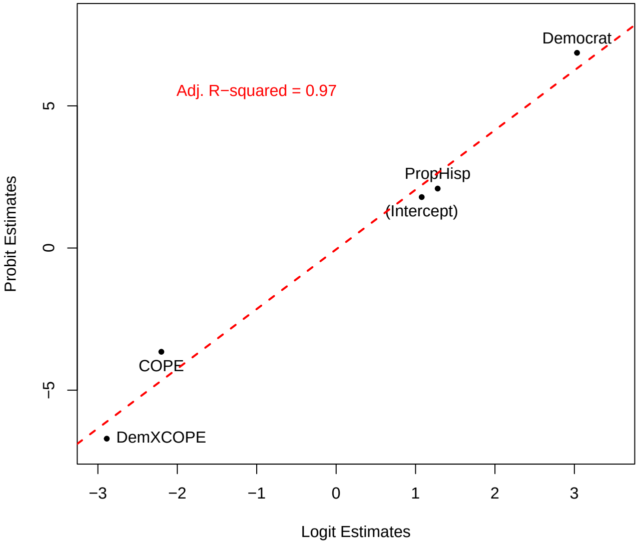
Coefficient Plots (via `modelplot`)



Faceted Plot (also via `modelplot`)



$\hat{\beta}_{\text{probit}}$ vs. $\hat{\beta}_{\text{logit}}$

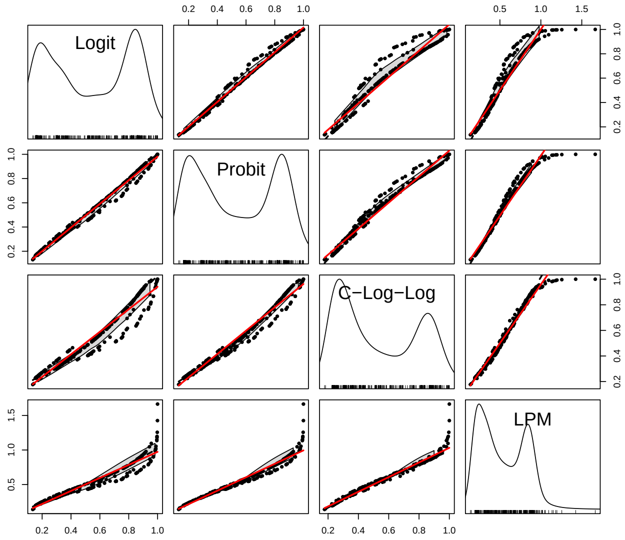


All The Models, Compared

Table 2: All The Models

	Logit	Probit	C-Log-Log	LPM
(Intercept)	1.792	1.078	0.701	0.861
	(0.275)	(0.153)	(0.133)	(0.042)
Proportion Hispanic	2.091	1.279	1.383	0.379
	(0.794)	(0.467)	(0.498)	(0.141)
Democrat	6.866	3.034	2.030	0.739
	(1.547)	(0.739)	(0.570)	(0.140)
COPE Score	-3.650	-2.201	-2.338	-0.721
	(0.760)	(0.440)	(0.487)	(0.129)
Democrat × COPE Score	-6.705	-2.888	-1.749	-0.685
	(1.820)	(0.903)	(0.804)	(0.197)
Num.Obs.	434	434	434	434
R2				0.314
R2 Adj.				0.308
AIC	446.8	451.1	465.5	475.5
BIC	467.2	471.4	485.9	500.0
Log.Lik.	-218.414	-220.532	-227.763	-231.770
F	26.622	30.723	27.889	49.167
RMSE	0.40	0.41	0.42	0.41

Let's Compare Predictions, Too



- `R` / `glm` reports "deviances":
  - "Residual" deviance $= 2(\ln L_S - \ln L_M)$
  - "Null" deviance $= 2(\ln L_S - \ln L_N)$
  - stored in `object$deviance` and `object$null.deviance`
- So:

$$
\begin{aligned}
LR_{\boldsymbol{\beta}=\mathbf{0}} &= 2(\ln L_M - \ln L_N) \\
&= \text{“Null” deviance} - \text{“Residual” deviance}
\end{aligned}
$$

Example:

```
> LLR<-NAFTA.fit$null.deviance - NAFTA.fit$deviance

> LLR
[1] 162

> pchisq(LLR,4,lower.tail=FALSE)
[1] 5.04e-34
```

Interpretation: "Signs-n-Significance"

For both logit and probit:

- $\hat{\beta}_k > 0 \;\leftrightarrow\; \frac{\partial \Pr(Y=1)}{\partial X_k} > 0$
- $\hat{\beta}_k < 0 \;\leftrightarrow\; \frac{\partial \Pr(Y=1)}{\partial X_k} < 0$
- $\frac{\hat{\beta}_k}{\hat{\sigma}_k} \sim N(0,1)$

Interactions:

$$\hat{\beta}_{\texttt{COPE|Democrat=1}} \equiv \hat{\phi}_{\texttt{COPE}} = \hat{\beta}_3 + \hat{\beta}_4$$

$$\text{s.e.}(\hat{\beta}_{\texttt{COPE|Democrat=1}}) = \sqrt{\text{Var}(\hat{\beta}_3) + (\texttt{Democrat})^2 \text{Var}(\hat{\beta}_4) + 2\,(\texttt{Democrat})\,\text{Cov}(\hat{\beta}_3, \hat{\beta}_4)}$$

Interactions ("by hand")

```
> NAFTA.fit$coeff[4]+NAFTA.fit$coeff[5]

 COPE
-10.4

> # z-statistic:
>
> (NAFTA.fit$coeff[4]+NAFTA.fit$coeff[5]) /
+   (sqrt(vcov(NAFTA.fit)[4,4] +
+   (1)^2*vcov(NAFTA.fit)[5,5] +
+   2*1*vcov(NAFTA.fit)[4,5]))
 COPE
-6.25

> # Square that, and it's a chi-square statistic:
>
> ((NAFTA.fit$coeff[4]+NAFTA.fit$coeff[5]) /
+   (sqrt(vcov(NAFTA.fit)[4,4] +
+           (1)^2*vcov(NAFTA.fit)[5,5] +
+           2*1*vcov(NAFTA.fit)[4,5])))^2
COPE
  39
```

(Or use car...)

```
> library(car)
> linearHypothesis(NAFTA.fit,"COPE+DemXCOPE=0")
Linear hypothesis test

Hypothesis:
COPE  + DemXCOPE = 0

Model 1: restricted model
Model 2: Vote ~ Democrat + PropHisp + COPE + DemXCOPE

  Res.Df Df Chisq    Pr(>Chisq)
1    430
2    429  1    39 0.00000000042 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

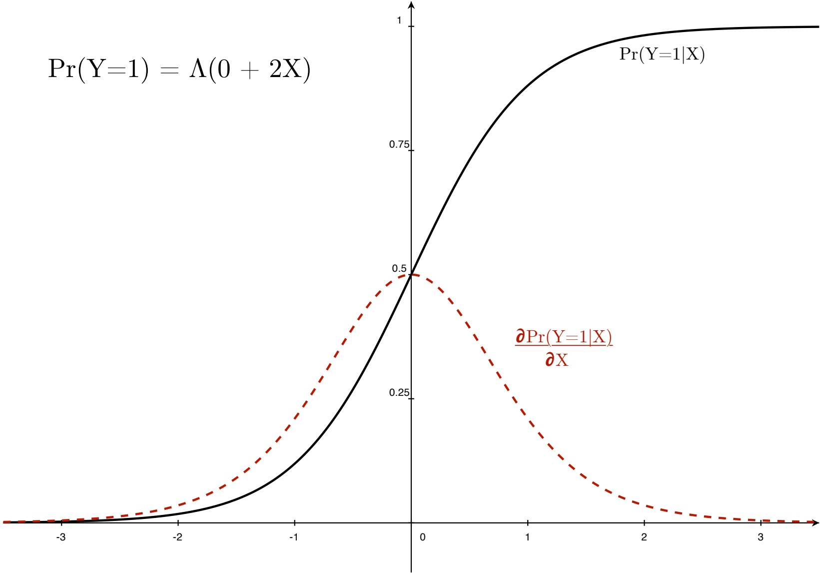
The *marginal effect* is:

$$
\begin{aligned}
\frac{\partial \Pr(\hat{Y}_i = 1)}{\partial X_k} &= \frac{\partial F(\mathbf{X}_i \hat{\boldsymbol{\beta}})}{\partial X_k} \\
&= f(\mathbf{X}_i \hat{\boldsymbol{\beta}})\hat{\beta}_k \\
&= \Lambda(\mathbf{X}_i \hat{\boldsymbol{\beta}})[1 - \Lambda(\mathbf{X}_i \hat{\boldsymbol{\beta}})]\hat{\beta}_k \quad \text{(for logit) or} \\
&= \phi(\mathbf{X}_i \hat{\boldsymbol{\beta}})\hat{\beta}_k \quad \text{(for probit)}
\end{aligned}
$$

Note that these depend on $\mathbf{X}\hat{\boldsymbol{\beta}}$, which means we either have to:

1. ...hold $\mathbf{X}\hat{\boldsymbol{\beta}}$ constant at some value(s), or
2. ...average over the actual values of $\mathbf{X}_i\hat{\boldsymbol{\beta}}$ observed in the data.

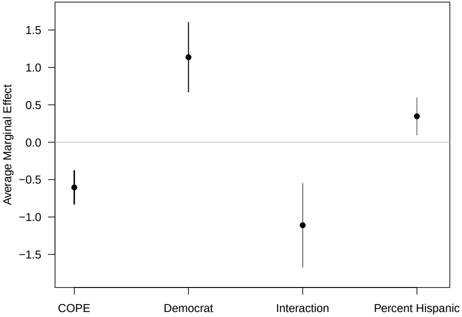
Marginal Effects Illustrated



Marginal Effects In Action

```
> summary(margins(NAFTA.fit))
   factor     AME     SE       z      p   lower   upper
     COPE -0.6043 0.1139 -5.3048 0.0000 -0.8275 -0.3810
 Democrat  1.1366 0.2370  4.7953 0.0000  0.6720  1.6011
 DemXCOPE -1.1101 0.2858 -3.8836 0.0001 -1.6703 -0.5498
 PropHisp  0.3462 0.1280  2.7054 0.0068  0.0954  0.5970
```

Plotted:



Log-Odds of $Y = 1$ are linear in $\mathbf{X}$:

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

That implies that:

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

OR for a one-unit change in X_k:

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

OR for a δ-unit change in X_k:

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

Also:

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

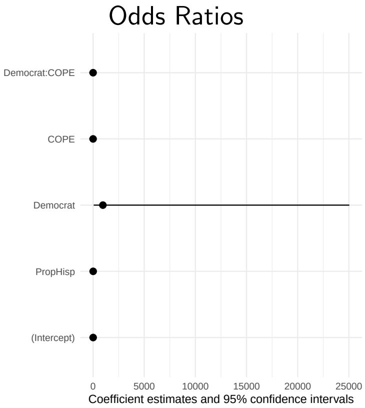
Odds Ratios Implemented

```
> P<-qnorm(0.975)

> lreg.or <- function(model)
+ {
+   coeffs <- coef(summary(model))
+   lowerCI <- exp(coeffs[ ,1] - P * coeffs[ ,2])
+   OR <- exp(coeffs[ ,1])
+   upperCI <- exp(coeffs[ ,1] + P * coeffs[ ,2])
+   lreg.or <- cbind(OR,lowerCI,upperCI)
+   lreg.or
+ }

> lreg.or(NAFTA.fit)
                   OR    lowerCI    upperCI
(Intercept)   5.99928  3.4965990    10.2933
PropHisp      8.09352  1.7068838    38.3770
Democrat    958.67832 46.1969511 19894.4757
COPE          0.02599  0.0058625     0.1152
DemXCOPE      0.00122  0.0000345     0.0434

> Or via -confint-...

> exp(cbind(OR=coef(NAFTA.fit),confint.default(NAFTA.fit)))
                   OR      2.5 %     97.5 %
(Intercept)   5.99928  3.4965990    10.2933
PropHisp      8.09352  1.7068838    38.3770
Democrat    958.67832 46.1969511 19894.4757
COPE          0.02599  0.0058625     0.1152
DemXCOPE      0.00122  0.0000345     0.0434
```

Odds Ratios via `modelsummary` / `modelplot`

Table 3: Odds Ratios

	(1)
(Intercept)	5.999
	(1.652)
PropHisp	8.094
	(6.427)
Democrat	958.678
	(1483.358)
COPE	0.026
	(0.020)
Democrat × COPE	0.001
	(0.002)
Num.Obs.	434
AIC	446.8
BIC	467.2
Log.Lik.	-218.414
F	26.622
RMSE	0.40



What Does This *Mean*?

```
> NAFTA.fit

Coefficients:
(Intercept)     PropHisp     Democrat         COPE     DemXCOPE
       1.79         2.09         6.87        -3.65        -6.71

> exp(cbind(OR=coef(NAFTA.fit),confint.default(NAFTA.fit)))
                   OR      2.5 %     97.5 %
(Intercept)   5.99928  3.4965990    10.2933
PropHisp      8.09352  1.7068838    38.3770
Democrat    958.67832 46.1969511 19894.4757
COPE          0.02599  0.0058625     0.1152
DemXCOPE      0.00122  0.0000345     0.0434
```

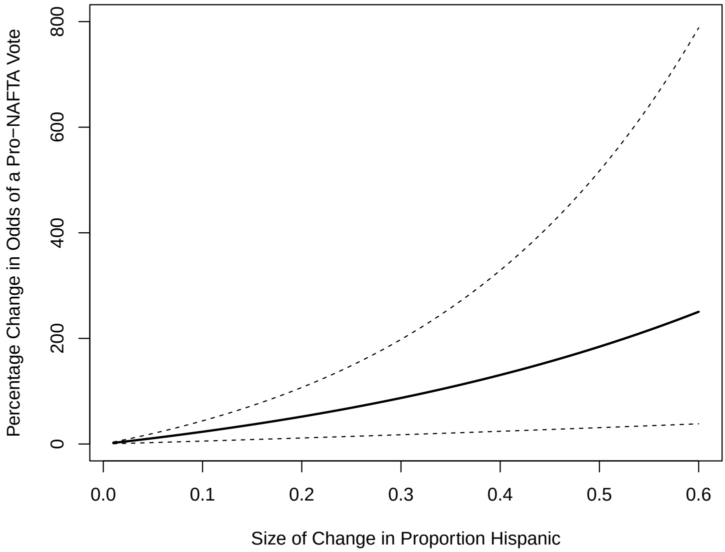
Consider `PropHispc`:

- A one-unit change (from 0 percent to 100 percent) in `PropHisp` corresponds to a $[(8.094 - 1) \times 100] =$ **709 percent expected increase in the odds** of a member of Congress voting in favor of NAFTA.
- A change of 0.10 (that is, a ten percentage-point increase) in the proportion of a member's district who is Hispanic corresponds to an **odds ratio** of:

$$\begin{aligned} \exp(2.09 \times 0.10) &= \exp(0.209) \\ &= 1.232 \end{aligned}$$

- This means that an increase of 0.10 in `PropHisp` corresponds to a $[(1.232 - 1) \times 100] =$ **23.2 percent expected increase in the odds** that a member of Congress would have voted in favor of NAFTA.
- For an increase of 0.20 (that is, 20 percentage points), the corresponding odds ratio and percent increase are 1.519 and 51.9 percent, respectively.

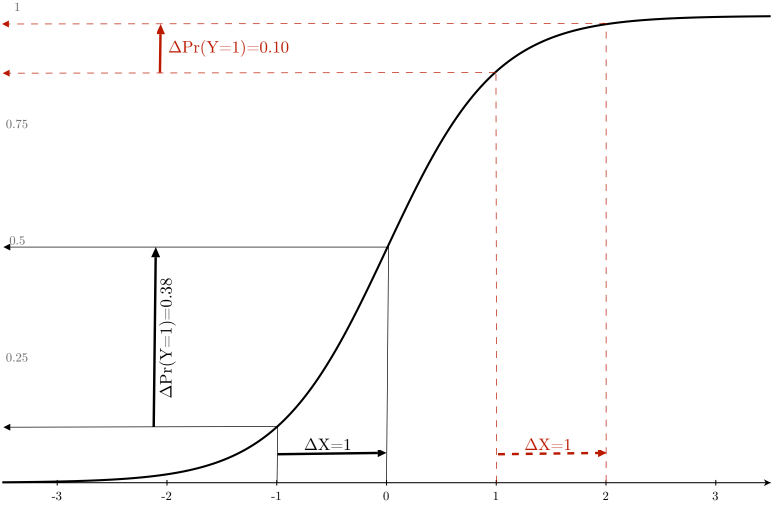
Percentage Change in Odds, by ΔPropHisp



Predicted probabilities:

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

Predicted Probabilities Illustrated



Predicted Probabilities: Standard Errors

The variability of a predicted probability is:

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

where $f(\cdot)$ refers to the standard logistic (for logit) or standard normal (for probit) density, and $F(\cdot)$ is the corredponding cumulative distribution function (CDF).

So,

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

Changes in $\widehat{\Pr(Y=1)}$:

$$
\begin{aligned}
\Delta\widehat{\Pr(Y=1)}_{\mathbf{X}_A \to \mathbf{X}_B} &= \frac{\exp(\mathbf{X}_B\hat{\beta})}{1+\exp(\mathbf{X}_B\hat{\beta})} - \frac{\exp(\mathbf{X}_A\hat{\beta})}{1+\exp(\mathbf{X}_A\hat{\beta})} \\
&\text{or} \\
&= \Phi(\mathbf{X}_B\hat{\beta}) - \Phi(\mathbf{X}_A\hat{\beta})
\end{aligned}
$$

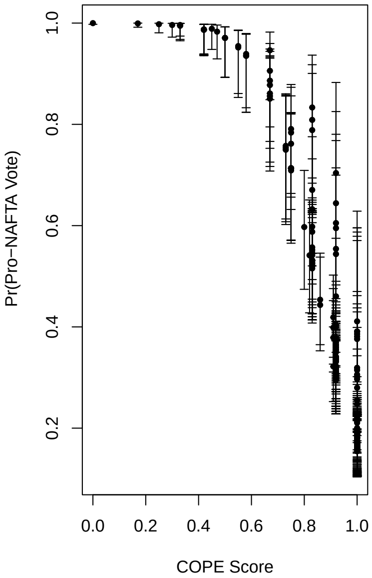
Standard errors for these changes are obtainable via delta method, bootstrap, etc...

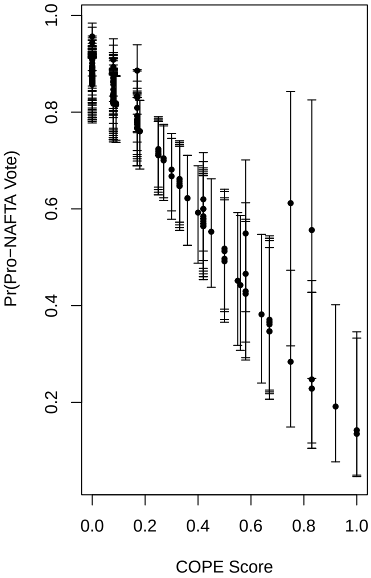
In-Sample Predictions

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

> hats<-predict(NAFTA.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 ...


> 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)

> with(NAFTA,
+  plotCI(COPE[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)"))

> with(NAFTA,
+  plotCI(COPE[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)"))
```

In-Sample Predictions





Out-of-Sample Predictions

"Fake" data:

```
> sim.data<-data.frame(PropHisp=mean(NAFTA$PropHisp),Democrat=rep(0:1,101),
                        COPE=seq(from=0,to=1,length.out=101))
> sim.data$DemXCOPE<-sim.data$Democrat*sim.data$COPE
```

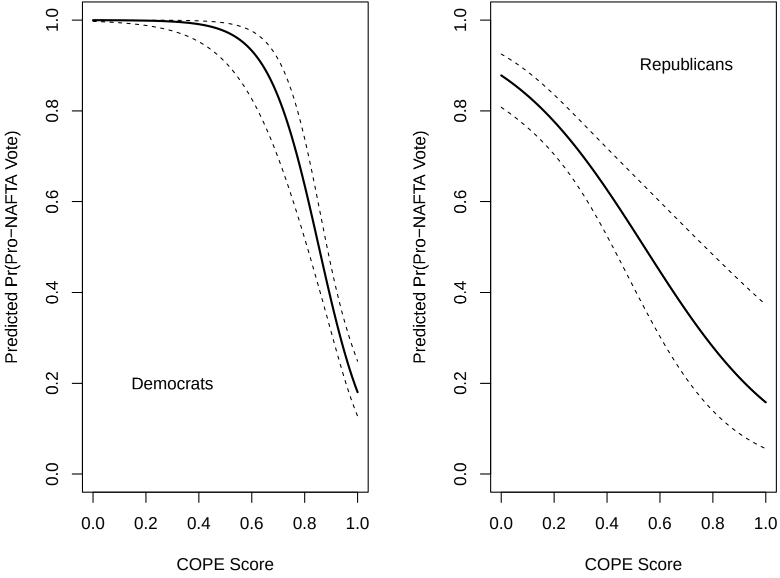
Generate predictions:

```
> OutHats<-predict(NAFTA.fit,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))
```

Plot:

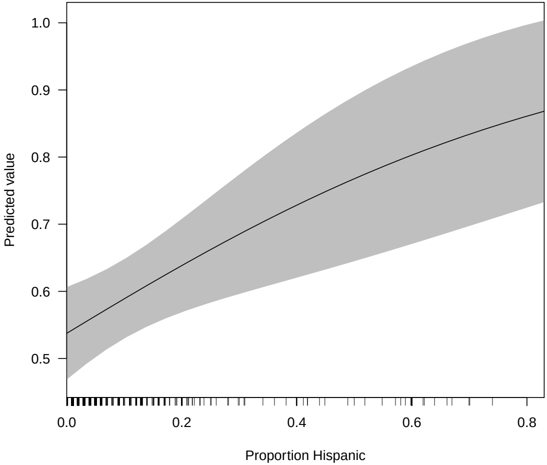
```
> both<-cbind(sim.data,OutHats)
> both<-both[order(both$COPE,both$Democrat),]
> bothD<-both[both$Democrat==1,]
> bothR<-both[both$Democrat==0,]

> par(mfrow=c(1,2))

> plot(bothD$COPE,bothD$fit,t="l",lwd=2,ylim=c(0,1),
     xlab="COPE Score",ylab="Predicted Pr(Pro-NAFTA Vote)")
> lines(bothD$COPE,bothD$OutHatsUB,lty=2)
> lines(bothD$COPE,bothD$OutHatsLB,lty=2)
> text(0.3,0.2,label="Democrats")

> plot(bothR$COPE,bothR$fit,t="l",lwd=2,ylim=c(0,1),
     xlab="COPE Score",ylab="Predicted Pr(Pro-NAFTA Vote)")
> lines(bothR$COPE,bothR$OutHatsUB,lty=2)
> lines(bothR$COPE,bothR$OutHatsLB,lty=2)
> text(0.7,0.9,label="Republicans")
```

Out-of-Sample Predictions



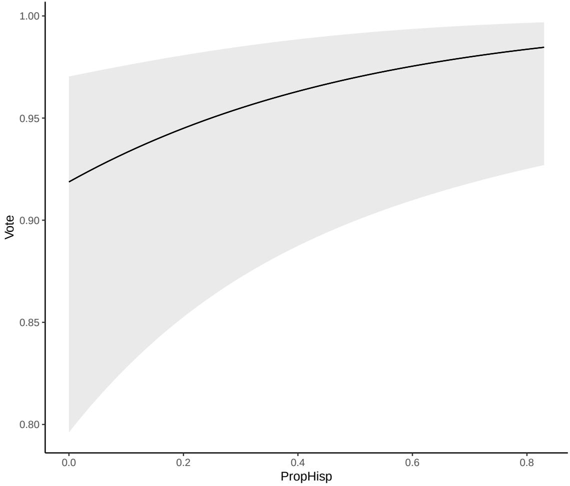
Single-Variable Example (using `cplot`)

```
> cplot(NAFTA.fit,"PropHisp",xlab="Proportion Hispanic")
```



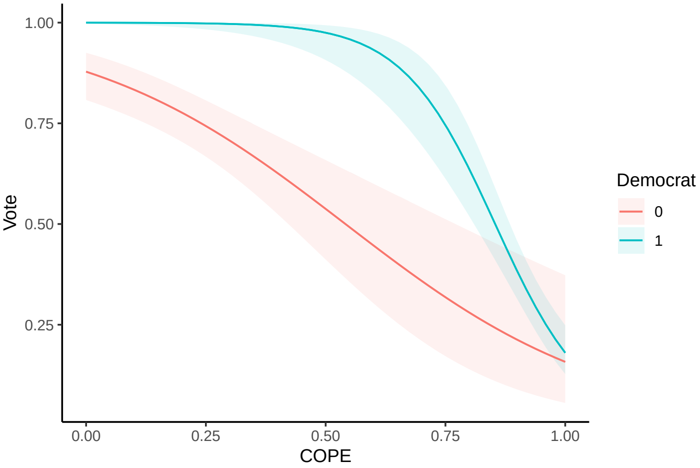
Similar, using `marginaleffects::plot_prediction`

```
> plot_predictions(NAFTA.fit,condition="PropHisp") + theme_classic()
```



Interactive part, using plot_prediction

```
> plot_predictions(fit,condition=c("COPE","Democrat")) + theme_classic()
```



Goodness of Fit

Some alternatives....

- Pseudo-R^2 (no!)
- Proportional reduction in error (PRE) – a/k/a "accuracy"
- ROC curves.

Model Fit: Predictions

Suppose we assign:

$$\hat{Y}_i = 0 \quad \text{if} \quad \widehat{\Pr(Y_i = 1)} \leq \tau$$
$$\hat{Y}_i = 1 \quad \text{if} \quad \widehat{\Pr(Y_i = 1)} > \tau$$

This would then give us a "confusion matrix":

	Predicted $\hat{Y}_i$	
Actual Y_i	$\hat{Y}_i = 0$	$\hat{Y}_i = 1$
$Y_i = 0$	True Negative ("TN")	False Positive ("FP")
$Y_i = 1$	False Negative ("FN")	True Positive ("TP")

This means we have:

- Total actual negatives = TN + FP
- Total actual positives = TP + FN
- Number correctly predicted = TP + TN

Proportional Reduction in Error (PRE):

$$\text{PRE} = \frac{N_{MC} - N_{NC}}{N - N_{NC}}$$

- N_{NC} = number correctly predicted under the "null model,"
- N_{MC} = number correctly predicted under the estimated model,
- N = total number of observations.

PRE tells us how much (proportionally) better our model does at predicting Y in-sample than would a model that only contained an intercept.

PRE: Example

```
> Assume tau = 0.5...
>
> table(NAFTA.fit$fitted.values>0.5,nafta$vote==1)

       FALSE TRUE
  FALSE   148   49
  TRUE     52  185
```

$$\begin{aligned} \text{PRE} &= \frac{N_{MC} - N_{NC}}{N - N_{NC}} \\ &= \frac{(148 + 185) - 234}{434 - 234} \\ &= \frac{99}{200} \\ &= \mathbf{0.495} \end{aligned}$$

Chi-Square test:

```
> chisq.test(NAFTA.fit$fitted.values>0.5,NAFTA$Vote==1)

 Pearson's Chi-squared test with Yates' continuity correction

data:  NAFTA.fit$fitted.values > 0.5 and NAFTA$Vote == 1
X-squared = 120, df = 1, p-value <2e-16
```

Related Ideas

Concepts:

- *Sensitivity* (or "true positive rate")
  - The proportion of all actual positives that were predicted correctly
  - Sensitivity $= \frac{\text{TP}}{\text{TP} + \text{FN}}$

- *Specificity* (or "true negative rate")
  - The proportion of all actual negatives that were predicted correctly
  - Specificity $= \frac{\text{TN}}{\text{TN} + \text{FP}}$

- False positive rate $= 1-$*Specificity*
- False negative rate $= 1-$*Sensitivity*

Varying τ

Suppose we set $\tau = 0.00001$. Then:

- We would essentially *always* predict $\hat{Y}_i = 1$, which means
- ...we would always correctly predict all the actual positives (maximize TPs), but
- ...we'd also always get every actual negative wrong (maximize FPs).

Similarly, if we set $\tau = 0.99999$. Then:

- We would essentially *always* predict $\hat{Y}_i = 0$, which means
- ...we would always correctly predict all the actual negatives (maximize TNs), but
- ...also always get every actual positive wrong (maximize FNs).

Values of τ between the extremes trade off true positives for false positives; as τ increases, we have fewer of the former and more of the latter.

NAFTA Examples

```
> # Tau = 0.2:

> Hats02<-ifelse(NAFTA.fit$fitted.values>0.2,1,0)
> CrossTable(NAFTA$Vote,Hats02,prop.r=FALSE,prop.c=FALSE,
             prop.t=FALSE,prop.chisq=FALSE)

             | Hats02
  NAFTA$Vote |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |        96 |       104 |       200 |
-------------|-----------|-----------|-----------|
           1 |         1 |       233 |       234 |
-------------|-----------|-----------|-----------|
Column Total |        97 |       337 |       434 |
-------------|-----------|-----------|-----------|

TPR = 233/234 = 0.996
FPR = 104/200 = 0.520


> # Tau = 0.8:

> Hats08<-ifelse(NAFTA.fit$fitted.values>0.8,1,0)
> CrossTable(NAFTA$Vote,Hats08,prop.r=FALSE,prop.c=FALSE,
             prop.t=FALSE,prop.chisq=FALSE)

             | Hats08
  NAFTA$Vote |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       178 |        22 |       200 |
-------------|-----------|-----------|-----------|
           1 |       123 |       111 |       234 |
-------------|-----------|-----------|-----------|
Column Total |       301 |       133 |       434 |
-------------|-----------|-----------|-----------|

TPR = 111/234 = 0.474
FPR = 178/200 = 0.890
```

"Receiver Operating Characteristic" (ROC) Curves

Now, imagine:

1. Fit a model
2. Choose a value of τ very near 0
3. Generate $\hat{Y}_i$s
4. Calculate and save the TPR and FPR for that value of τ
5. Increase τ by a very small amount
6. Go to (3), and repeat until τ is very close to 1.0

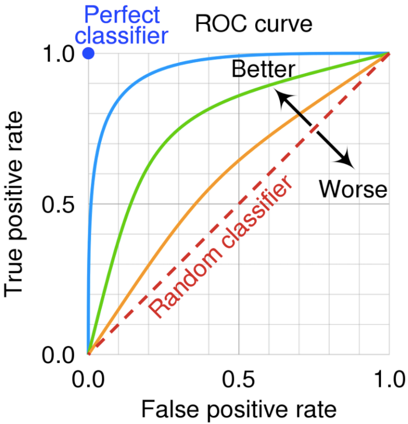
We could then plot the true positive rate vs. false positive rate (i.e., *Specificity* vs. 1 - *Sensitivity*)

ROC Curves (continued)

So:

- If the model fits perfectly, it will have a 1.0 true positive rate, and a 0.0 false negative rate
- If the model fits no better than random chance, the curve defined by those points will be a diagonal line.
- (Intuition: If each prediction is no better than a (weighted) coin flip, the rate of true positives and false positives will increase together.)
- In between these extremes, better-fitting models will have curves that are closer to the upper-left corner

"AUROC" (or "AUC"): Area under the ROC curve → assessment of model fit



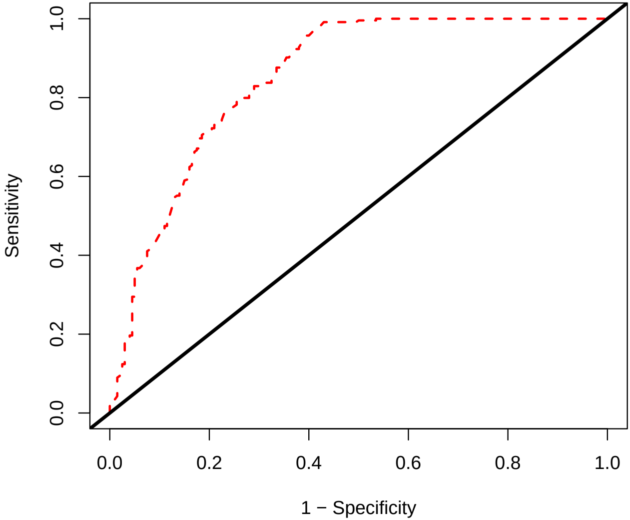
(Source)

ROC Curves Implemented

Code:

```
> library(ROCR)

> NAFTA.hats<-predict(NAFTA.fit,type="response")

> preds<-ROCR::prediction(NAFTA.hats,NAFTA$Vote)

> plot(ROCR::performance(preds,"tpr","fpr"),lwd=2,lty=2,
       col="red",xlab="1 - Specificity",ylab="Sensitivity")

> abline(a=0,b=1,lwd=3)
```

ROC Curve: Example



Interpreting AUROC Curves

- Area under ROC $=$ 0.90-1.00 $\rightarrow$ Excellent (A)
- Area under ROC $=$ 0.80-0.90 $\rightarrow$ Good (B)
- Area under ROC $=$ 0.70-0.80 $\rightarrow$ Fair (C)
- Area under ROC $=$ 0.60-0.70 $\rightarrow$ Poor (D)
- Area under ROC $=$ 0.50-0.60 $\rightarrow$ Total Failure (F)

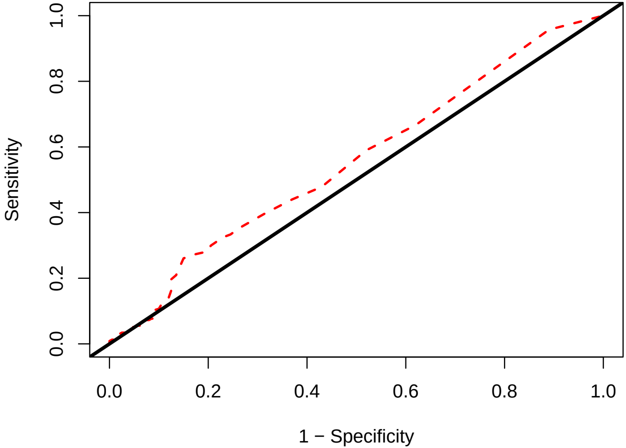
ROC Curve: A Poorly-Fitting Model

Model is:

$$\Pr(\texttt{vote}_i = 1) \quad = \quad f[\beta_0 + \beta_1(\texttt{PropHisp}_i) + u_i]$$

```
> NAFTA.bad<-with(NAFTA,
                  glm(Vote~PropHisp,family=binomial(link="logit")))
> NAFTA.bad.hats<-predict(NAFTA.bad,type="response")
> bad.preds<-ROCR::prediction(NAFTA.bad.hats,NAFTA$Vote)

> plot(ROCR::performance(bad.preds,"tpr","fpr"),lwd=2,lty=2,
       col="red",xlab="1 - Specificity",ylab="Sensitivity")
> abline(a=0,b=1,lwd=3)
```

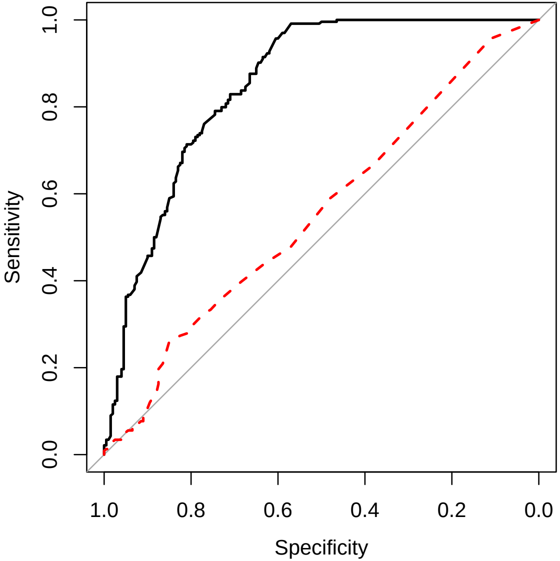
Bad ROC!



Comparing ROCs

```
> install.packages("pROC")
> library(pROC)

> GoodROC<-roc(NAFTA$Vote,NAFTA.hats,ci=TRUE)
> GoodAUC<-auc(GoodROC)
> BadROC<-roc(NAFTA$Vote,NAFTA.bad.hats,ci=TRUE)
> BadAUC<-auc(BadROC)

> GoodAUC
Area under the curve: 0.85

> BadAUC
Area under the curve: 0.556
```

Combined Plot



Useful Things

Model Fitting, etc.:

- `glm` (in base `stats`)
  - Binary responses = `family(binomial)`
  - Links: `logit`, `probit`, `cloglog`, `log`, `cauchit` (Cauchy)
- Some `easystats` packages:
  - `datawizard` (standardizing variables, etc.)
  - `correlation` (what the name says...)

Model Interpretation + Visualization:

- `modelsummary` (tables and plots of estimates, ORs, etc.)
- `marginaleffects` (generate and plot of predictions, etc.)
- `margins` (marginal effects)
- `ROCR`, `pROC` (generate / plot ROC curves, calculate AUROC)
- `easystats` packages:
  - `report` + `parameters` (tables, output, etc.)
  - `modelbased` + `effectsize` (substantive interpretation of models)
  - `performance` (model fit: R^2, AUROC, etc.)

Models for Event Counts

Things that are not counts:

- Ordinal scales/variables
- Grouped Binary Data
  - $\frac{N \text{ of ``successes''}}{N \text{ of ``trials''}}$
  - Binomial data
  - $=$ counts only if Pr("success") is small

Count properties:

- Discrete / integer-valued
- Non-negative
- "Cumulative"

Count Data: Motivation

Events:

$$\text{Arrival Rate} = \lambda$$

$$\Pr(\text{Event})_{t,t+h} = \lambda h$$

$$\Pr(\text{No Event})_{t,t+h} = 1 - \lambda h$$

Count of events:

$$\begin{aligned} \Pr(Y_t = y) &= \frac{\exp(-\lambda h)\lambda h^y}{y!} \\ &= \frac{\exp(-\lambda)\lambda^y}{y!} \end{aligned}$$

Poisson Assumptions

Three assumptions:

- No Simultaneous Events
- Constant Arrival Rate
- Independent Event Arrivals

For M independent Bernoulli trials with (sufficiently small) probability of success π and where $M\pi \equiv \lambda > 0$,

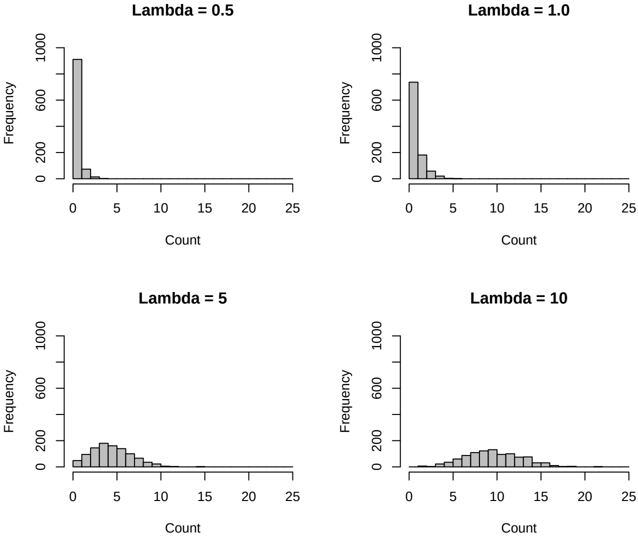
$$\begin{aligned}
\Pr(Y_i = y) &= \lim_{M\to\infty}\left[\binom{M}{y}\left(\frac{\lambda}{M}\right)^{y}\left(1-\frac{\lambda}{M}\right)^{M-y}\right] \\
&= \frac{\lambda^{y}\exp(-\lambda)}{y!}
\end{aligned}$$

Poisson: Characteristics

A Poisson variate:

- Discrete
- $\mathrm{E}(Y) = \mathrm{Var}(Y) = \lambda$
- Is not preserved under affine transformations...
- For $X \sim \text{Poisson}(\lambda_X)$ and $Y \sim \text{Poisson}(\lambda_Y)$,
  $Z = X + Y \sim \text{Poisson}(\lambda_{X+Y})$ *iff* X and Y are *independent* but
- ...same is not true for differences.
- $\lambda \to \infty \iff Y \sim N$

Poissons: Examples



Suppose

$$E(Y_i) \equiv \lambda_i = \exp(\mathbf{X}_i\beta)$$

then

$$\Pr(Y_i = y|\mathbf{X}_i, \beta) = \frac{\exp[-\exp(\mathbf{X}_i\beta)][\exp(\mathbf{X}_i\beta)]^y}{y!}$$

Likelihood:

$$L = \prod_{i=1}^{N} \frac{\exp[-\exp(\mathbf{X}_i\boldsymbol{\beta})][\exp(\mathbf{X}_i\boldsymbol{\beta})]^{Y_i}}{Y_i!}$$

$\rightarrow$ Log-likelihood:

$$\ln L = \sum_{i=1}^{N}[-\exp(\mathbf{X}_i\boldsymbol{\beta}) + Y_i\mathbf{X}_i\boldsymbol{\beta} - \ln(Y_i!)]$$

Example: Federal Judicial Review, 1789-2021

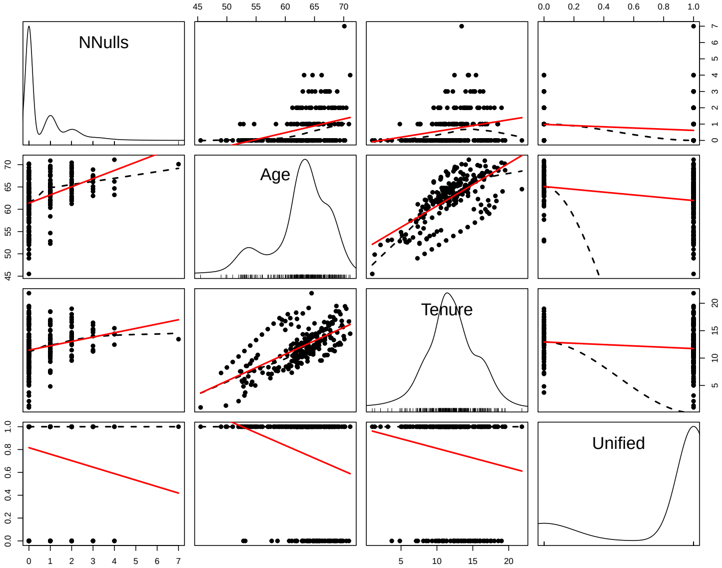
Data:

- Y_i = number of Acts of Congress overturned by the Supreme Court in each year (`NNulls`)

- Predictors:
  - The *mean age* (`Age`) of the Supreme Court's justices ($\bar{X} = 62.6, \sigma = 5, \mathrm{E}(\hat{\beta}) > 0$)
  - The *mean tenure* (`Tenure`) of the Supreme Court's justices ($\bar{X} = 12.0, \sigma = 3.5, \mathrm{E}(\hat{\beta}) > 0$)
  - Whether (1) or not (0) there was *unified government* (`Unified`) ($\bar{X} = 0.78, \mathrm{E}(\hat{\beta}) < 0$)

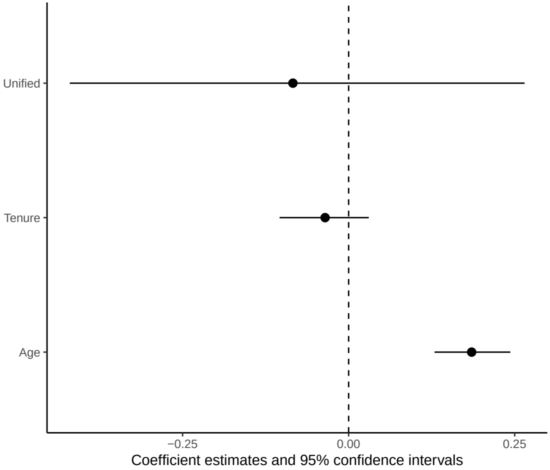
```
> psych::describe(NewDahl,fast=TRUE,skew=TRUE)
                vars   n    mean    sd median    min    max range  skew kurtosis   se
Year               1 233 1905.00 67.41 1905.0 1789.0 2021.0 232.0  0.00    -1.22 4.42
NConstDecisions    2 233   17.96 19.11   12.0    0.0   85.0  85.0  1.38     1.48 1.25
NNulls             3 233    0.70  1.06    0.0    0.0    7.0   7.0  1.96     5.39 0.07
Age                4 233   62.65  4.96   63.6   45.5   71.1  25.6 -0.84     0.29 0.32
Tenure             5 233   12.00  3.54   11.9    1.0   21.8  20.8 -0.19     0.25 0.23
Unified            6 233    0.78  0.42    1.0    0.0    1.0   1.0 -1.32    -0.26 0.03
```

Federal Judicial Review, 1789-2021



```
> nulls.poisson<-glm(NNulls~Age+Tenure+Unified,family="poisson",
+                    data=NewDahl)
> summary(nulls.poisson)

Call:
glm(formula = NNulls ~ Age + Tenure + Unified, family = "poisson",
    data = NewDahl)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.7638     1.6993   -6.92  4.4e-12 ***
Age           0.1852     0.0291    6.36  2.0e-10 ***
Tenure       -0.0354     0.0343   -1.03     0.30
Unified      -0.0839     0.1743   -0.48     0.63
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 339.50  on 232  degrees of freedom
Residual deviance: 266.69  on 229  degrees of freedom
AIC: 497.7

Number of Fisher Scoring iterations: 5
```

Coefficient Plot (using `modelplot`)



“Exposure” and “Offsets”

If we relax the assumption of equal “exposure,” we get:

$$\mathrm{E}(Y_i|\mathbf{X}_i, M_i) = \lambda_i M_i$$

i.e., the expected number of events is proportional to *exposure* M_i.

Note that now, instead of:

$$\ln[E(Y_i)] = \mathbf{X}_i\boldsymbol{\beta}$$

we have:

$$\ln\left[E\left(\frac{Y_i}{M_i}\right)\right] = \mathbf{X}_i\boldsymbol{\beta}$$

which is a *rate*, and the same as:

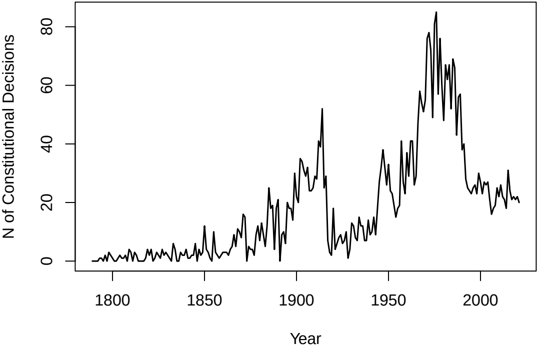
$$\ln[E(Y_i)] = \ln(M_i) + \mathbf{X}_i\boldsymbol{\beta}$$

that is, including $\ln(M_i)$ in $\mathbf{X}$ with $\beta_{\ln(M)} = 1$.

Exposure Example

For the judicial review (1789-2021) data:

- SCOTUS (typically) reviews many constitutional cases per year
- The number of such cases is the *possible* number of nullifications



Adding an "offset":

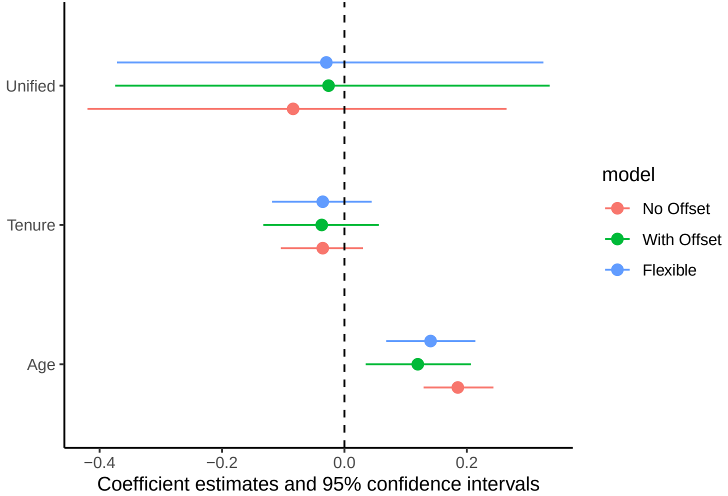
```
> nulls.poisson2<-glm(NNulls~Age+Tenure+Unified,family="poisson",
+                    offset=log(NConstDecisions+1),data=NewDahl)
> summary(nulls.poisson2)

Call:
glm(formula = NNulls ~ Age + Tenure + Unified, family = "poisson",
    data = NewDahl, offset = log(NConstDecisions + 1))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.6120     2.4552   -4.32 0.000015 ***
Age           0.1199     0.0440    2.72   0.0065 **
Tenure       -0.0371     0.0483   -0.77   0.4420
Unified      -0.0259     0.1808   -0.14   0.8859
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 257.47  on 232  degrees of freedom
Residual deviance: 242.37  on 229  degrees of freedom
AIC: 473.3

Number of Fisher Scoring iterations: 6
```

Correcting for Exposure (continued)

Including the "offset" as a control:

```
> nulls.poisson3<-glm(NNulls~Age+Tenure+Unified+log(NConstDecisions+1),
+                     family="poisson",data=NewDahl)
> summary(nulls.poisson3)

Coefficients:
                         Estimate Std. Error z value      Pr(>|z|)
(Intercept)              -10.5838     2.0821   -5.08 0.00000037124 ***
Age                        0.1408     0.0371    3.79       0.00015 ***
Tenure                    -0.0354     0.0415   -0.85       0.39295
Unified                   -0.0296     0.1774   -0.17       0.86739
log(NConstDecisions + 1)   0.5744     0.0924    6.22 0.00000000051 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 339.50  on 232  degrees of freedom
Residual deviance: 223.14  on 228  degrees of freedom
AIC: 456.1

Number of Fisher Scoring iterations: 5


> # Wald test for b = 1.0:
>
> wald.test(b=coef(nulls.poisson3),Sigma=vcov(nulls.poisson3),Terms=4,H0=1)
Wald test:
----------

Chi-squared test:
X2 = 33.7, df = 1, P(> X2) = 0.0000000064
```

Model Comparisons



Interpretation: Incidence Rate Ratios

IRRs:

$$\frac{\hat{\lambda}|X_D = 1}{\hat{\lambda}|X_D = 0} = \frac{\exp(\hat{\beta}_0 + \bar{\mathbf{X}}\hat{\boldsymbol{\beta}} + (X_D = 1)\hat{\beta}_{X_D})}{\exp(\hat{\beta}_0 + \bar{\mathbf{X}}\hat{\boldsymbol{\beta}} + (X_D = 0)\hat{\beta}_{X_D})}$$
$$= \exp(\hat{\beta}_{X_D})$$

- Like ORs
- `Age`: IRR=exp(0.19) = 1.21

For a δ-unit change in X_k:

$$\text{IRR}_{X_k, X_{k+\delta}} = \exp(\delta \hat{\beta}_k)$$

So, a for ten-year difference in `Age`:

$$\begin{aligned} \text{IRR} &= \exp(10 \times 0.190) \\ &= \exp(1.90) \\ &= 6.69 \end{aligned}$$

Incidence Rate Ratios

Via mfx:

```
> library(mfx)
> nulls.poisson.IRR<-poissonirr(NNulls~Age+Tenure+Unified,
+                                data=NewDahl)

> nulls.poisson.IRR
Call:
poissonirr(formula = NNulls ~ Age + Tenure + Unified, data = NewDahl)

Incidence-Rate Ratio:
           IRR Std. Err.     z P>|z|
Age     1.2035    0.0350  6.36 2e-10 ***
Tenure  0.9652    0.0331 -1.03  0.30
Unified 0.9195    0.1603 -0.48  0.63
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Predicted Values ($\hat{Y}$s)

Mean predicted Y:

$$E(Y|\bar{\mathbf{X}}_i) = \exp[\bar{\mathbf{X}}_i\hat{\beta}]$$

In-Sample:

- R : in `$fitted.values` (or use `predict`)
- Stata : use `predict`

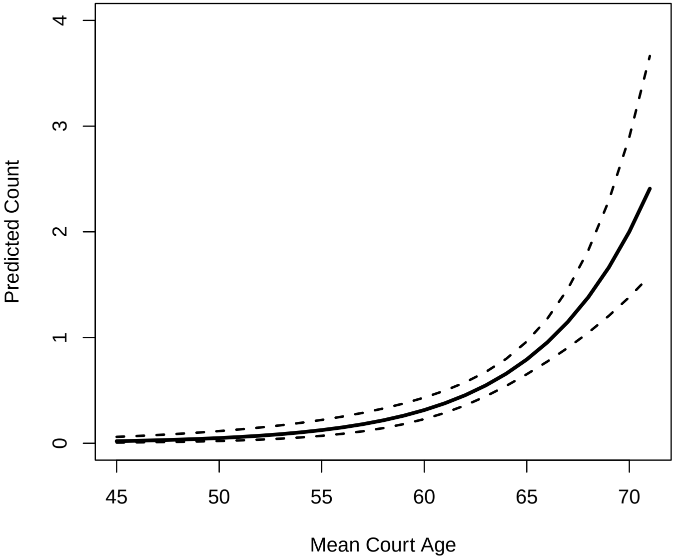
Out-of-Sample: use `predict`

Example: Out-Of-Sample Predicted Values

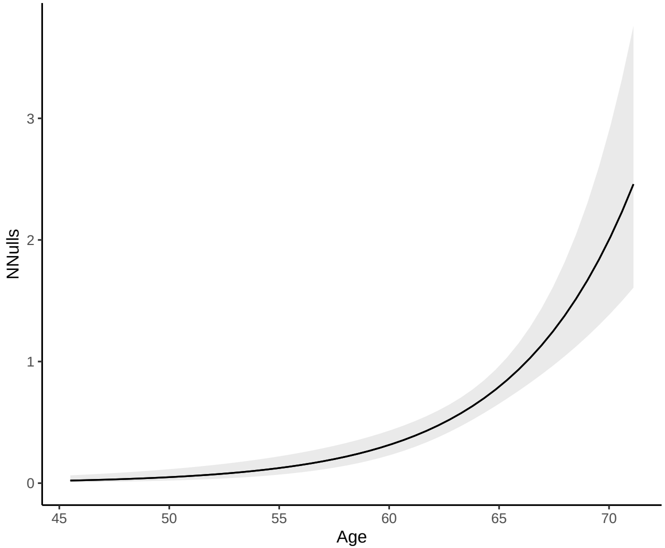
"By-hand" example:

```
> simdata<-data.frame(Age=seq(from=45,to=71,by=1),
+                     Tenure=mean(NewDahl$Tenure,na.rm=TRUE),
+                     Unified=1)
> nullhats<-predict(nulls.poisson,newdata=simdata,se.fit=TRUE)

> # NOTE: These are XBs, not predicted counts.
> # Transforming:

> nullhats$Yhat<-exp(nullhats$fit)
> nullhats$UB<-exp(nullhats$fit + 1.96*(nullhats$se.fit))
> nullhats$LB<-exp(nullhats$fit - 1.96*(nullhats$se.fit))
>
> plot(simdata$Age,nullhats$Yhat,t="l",lwd=3,ylim=c(0,5),ylab=
+        "Predicted Count", xlab="Mean Court Age")
> lines(simdata$Age,nullhats$UB,lwd=2,lty=2)
> lines(simdata$Age,nullhats$LB,lwd=2,lty=2)
```

Plotting Out-Of-Sample Predicted Values



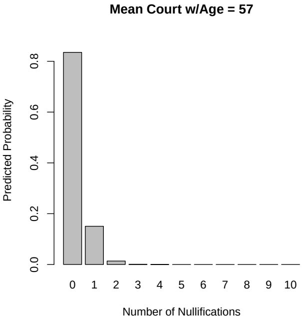
Same, Using `plot_predictions`



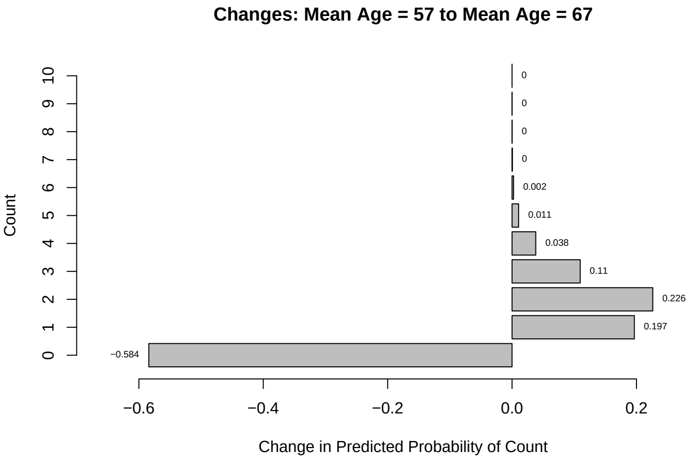
Predicted Probabilities

The predicted probability that $Y_i = y$ is:

$$\Pr(\widehat{Y_i = y}|\mathbf{X}_i, \hat{\beta}) = \frac{\exp[-\exp(\mathbf{X}_i\hat{\beta})][\exp(\mathbf{X}_i\hat{\beta})]^y}{y!}$$



Changes in Predicted Probabilities



More Count Model Things

Variations include:

- Models for Overdispersed and Underdispersed Counts
- Models where Over- / Underdispersion $= f(\mathbf{Z}_i\gamma)$
- Models for Censored / Truncated Counts
- "Zero-Inflated" and "Hurdle" Count Models
- Time Series of Counts
- Panel Data Models for Event Counts
- Spatial Count Models
- etc...