POLSCI 9592

Lecture 4: Model Fit and Evaluation

Dave Armstrong

Goals for This Session

1. Absolute Measures of Model Fit
2. Comparative Measures of Model Fit
3. Nested and Non-nested Model Tests
4. Model Specification Tests
5. Model Diagnostics
6. (Quasi-)Separation

Model Fit, Evaluation and Comparison

Now that we know how to interpret what is going on in these models, we need to develop a sense of how well they fit. We can do this through a number of different means.

- Likelihood-based fit measures
- Pseudo- R^2
- Information criteria
- Classification-based measures.
- Cross-validation.
- Specification tests.

Pseudo R-squared

Pseudo- R^2 measures rely on analogues to the linear model.

- Many different types, each of which can produce substantively different results from the others.
- If you use these measures, identify which one(s) you are using.
- None can be interpreted as the proportion of variation in the dependent variable explained by the independent variables.

McFadden's R-squared

McFadden's R^2 uses the following analogue to the linear model $R^2 = 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}}$. In this case,

$$\text{McFadden's } R^2 = 1 - \frac{\log \hat{L}(M_{\text{Full}})}{\log \hat{L}(M_{\text{Null}})}$$

There is also an adjustment to McFadden's R^2 to account for degrees of freedom:

$$\text{McFadden's } \bar{R}^2 = 1 - \frac{\log \hat{L}(M_{\text{Full}}) - K^*}{\log \hat{L}(M_{\text{Null}})}$$

where K^* is the number of parameters (not independent variables) in the model.

Cox & Snell and Cragg & Uhler R-squared

The Cox & Snell (or ML) R^2 uses the same analogy as McFadden's:

$$\text{Cox \& Snell } R^2 = 1 - \left\{ \frac{\log \hat{L}(M_{\text{Null}})}{\log \hat{L}(M_{\text{Full}})} \right\}^{\frac{2}{N}}$$

The Cox & Snell measure reaches a maximum of $1 - L(M_{\text{Null}})^{\frac{2}{N}}$, so Cragg and Uhler's R^2 norms it to reach a maximum at 1

$$\text{Cragg \& Uhler's } R^2 = \frac{1 - \left\{ \frac{\log \hat{L}(M_{\text{Null}})}{\log \hat{L}(M_{\text{Full}})} \right\}^{\frac{2}{N}}}{1 - L(M_{\text{Null}})^{\frac{2}{N}}}$$

Efron's R-squared

Efron's analogy to the linear model formula mentioned above is even more explicit.

$$\text{Efron's } R^2 = 1 - \frac{\sum_N (y_i - \hat{\pi}_i)^2}{\sum_N (y_i - \bar{y})^2}$$

McKelvey & Zavoina's R-squared

For models that can be defined in terms of a latent variable y^*: $y^* = \mathbf{x}\beta + \varepsilon$ where $\widehat{\mathrm{Var}}(\hat{y}^*) = \hat{\beta}' \widehat{\mathrm{Var}}(\mathbf{x}) \hat{\beta}$, The M&Z R^2 is:

$$\text{M\&Z } R^2 = \frac{\widehat{\mathrm{Var}}(\hat{y}^*)}{\widehat{\mathrm{Var}}(y^*)} = \frac{\widehat{\mathrm{Var}}(\hat{y}^*)}{\widehat{\mathrm{Var}}(y^*) + \mathrm{Var}(\varepsilon)}$$

Here, $\mathrm{Var}(\varepsilon)$ identifies the model. In logit it is $\frac{\pi^2}{3}$ and in the probit model it is 1.

Measures and Models

	OLS	Bianry	Ordered	Multinomial	Count
Log-Likelihood	✓	✓	✓	✓	✓
LR χ^2	✓	✓	✓	✓	✓
Information Criteria	✓	✓	✓	✓	✓
R^2 and $\tilde{R}^2$	✓				
Efron's R^2 and Tjur's D		✓			
McFadden, ML, C&U R^2		✓	✓	✓	✓
Count and Adjusted Count R^2		✓	✓	✓	
$Var(e)$, $Var(y^*)$, M&Z R^2		✓	✓		
PRE and ePRE		✓	✓	✓	
AIC	✓	✓	✓	✓	✓
BIC	✓	✓	✓	✓	✓

Example

```
library(DAMisc)
load("data/anes_2008_binary.rda")
dat$race <- rio::factorize(dat$race)
mod <- glm(voted ~ age + educ + income + poly(leftright, 2) +
    female + race, data=dat, family=binomial(link="logit"))
binfit(mod)
```

```
##                     Names1    vals1                        Names2    vals2
## 1 Log-Lik Intercept Only: -314.550          Log-Lik Full Model: -249.803
## 2                 D(575):  499.606                       LR(8):  129.495
## 3                                                  Prob > LR:    0.000
## 4          McFadden's R2:    0.206           McFadden's Adk R2:    0.177
## 5      ML (Cox-Snell) R2:    0.199 Cragg-Uhler (Nagelkerke) R2:    0.302
## 6  McKelvey & Zavoina R2:    0.357                  Efron's R2:    0.214
## 7               Count R2:    0.795              Adj Count R2:    0.104
## 8                    BIC:  556.935                       AIC:  517.606
```

Classification-based Measures

Proportional Reduction in Error. (PRE) tells us how much better we do at predicting y_i using a model versus guessing.

$$PRE = \frac{PCP - PMC}{1 - PMC}$$

Here,

$$\hat{y}_i = \begin{cases} 1 & \text{if } \hat{p}_i > 0.5 \\ 0 & \text{if } \hat{p}_i \leq .5 \end{cases}$$

$$PMC = max\left(\frac{\sum_i y_i}{N}, \frac{\sum_i (1 - y_i)}{N}\right)$$

$$PCP = \frac{\#(y_i = \hat{y}_i)}{N}$$

Expected PRE

Herron (1999) shows that there are lots of different arrangements of predicted probabilities that would generate the same PRE: For example, consider the two different sets of predictions:

y_i	$\hat{p}_i$	$\hat{y}_i$
1	0.60	1
0	0.51	1
1	0.55	1
1	0.75	1
0	0.25	0
0	0.45	0
0	0.40	0

$$PMC = 4/7 = 0.57$$
$$PCP = 6/7 = 0.86$$

y_i	$\hat{p}_i$	$\hat{y}_i$
1	0.80	1
0	0.51	1
1	0.65	1
1	0.95	1
0	0.15	0
0	0.25	0
0	0.10	0

$$PMC = 4/7 = 0.57$$
$$PCP = 6/7 = 0.86$$

ePRE

The ePRE is defined as:

$$ePRE = \frac{ePCP - ePMC}{1 - ePMC}$$

where:

$$ePMC = \bar{y}$$

$$ePCP = \frac{1}{N} \sum_i (y_i \hat{p}_i + (1 - y_i)(1 - \hat{p}_i))$$

Expected PRE Example

y_i	$\hat{p}_i$	$\hat{y}_i$
1	0.60	1
0	0.51	1
1	0.55	1
1	0.75	1
0	0.25	0
0	0.45	0
0	0.40	0

$$ePMC = 0.429$$

$$ePCP = 0.613$$

$$ePRE = 0.323$$

y_i	$\hat{p}_i$	$\hat{y}_i$
1	0.80	1
0	0.51	1
1	0.65	1
1	0.95	1
0	0.15	0
0	0.25	0
0	0.10	0

$$ePMC = 0.429$$

$$ePCP = 0.77$$

$$ePRE = 0.598$$

PRE in R

```
pre(mod, sim=T)
```

```
## mod1:  voted ~ age + educ + income + poly(leftright, 2) + female + race
## mod2:  voted ~ 1
##
## Analytical Results
##  PMC =  0.771
##  PCP =  0.795
##  PRE =  0.104
## ePMC =  0.646
## ePCP =  0.724
## ePRE =  0.220
##
## Simulated Results
##      median lower upper
##  PRE 0.104  0.052 0.157
## ePRE 0.215  0.152 0.269
```

Others...

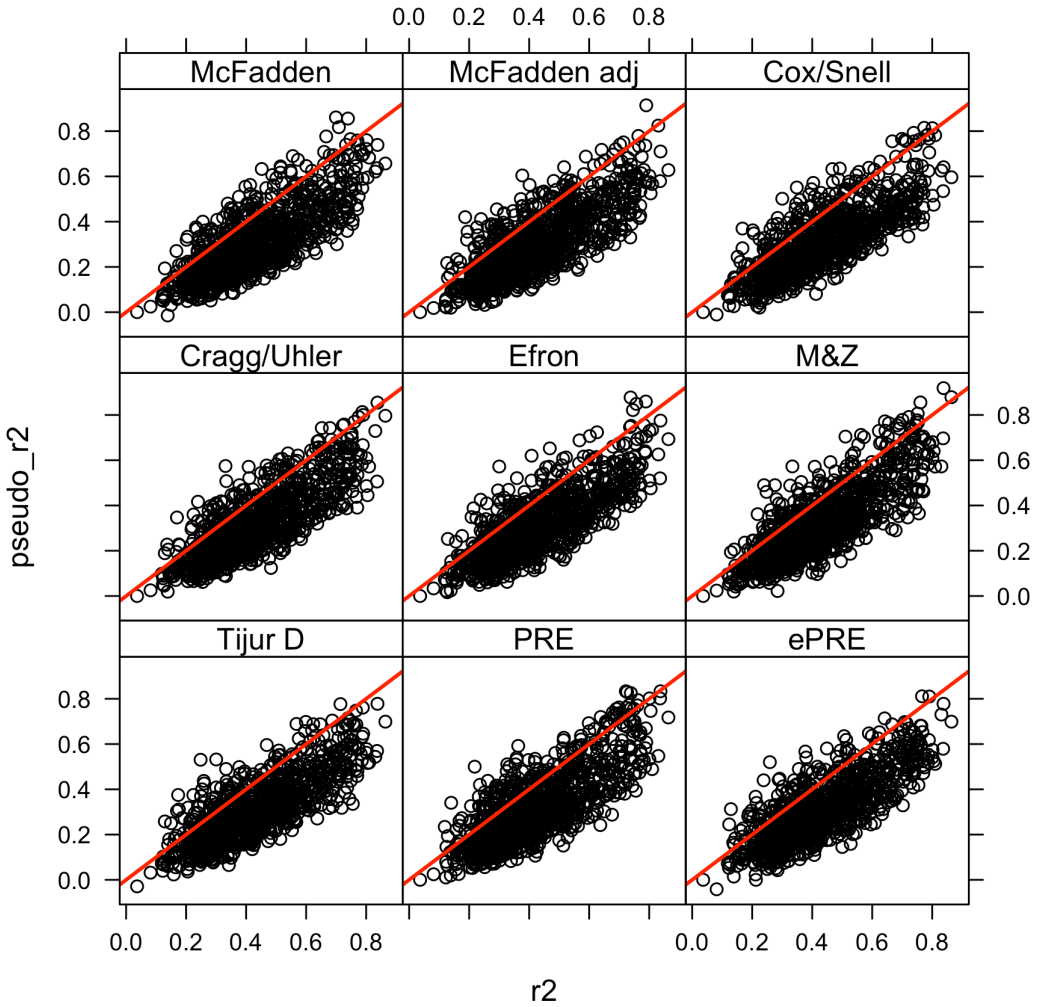
There are three other methods that consider the $Pr(Y = 1|Y = 1, X)$ and $1 - Pr(Y = 1|Y = 0, X)$.

- Separation Plot (Greenhill, Ward and Sacks, 2011) is a visual method that plots a single tick for each observation (where ones and zeros are different colors) sorted by $Pr(Y = 1|X)$.
- Tjur's D is a statistic that is defined as

$$\frac{1}{n_1} \sum_{y=1} Pr(Y = 1|X) - \frac{1}{n_0} \sum_{y=0} Pr(Y = 1|X)$$

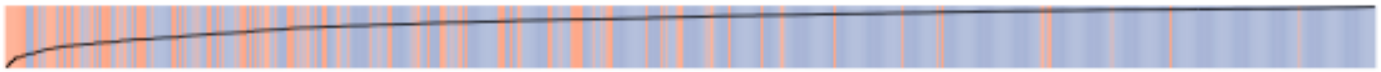
- Cross-validation

Comparison of Pseudo R-squared Measures



Separation Plot

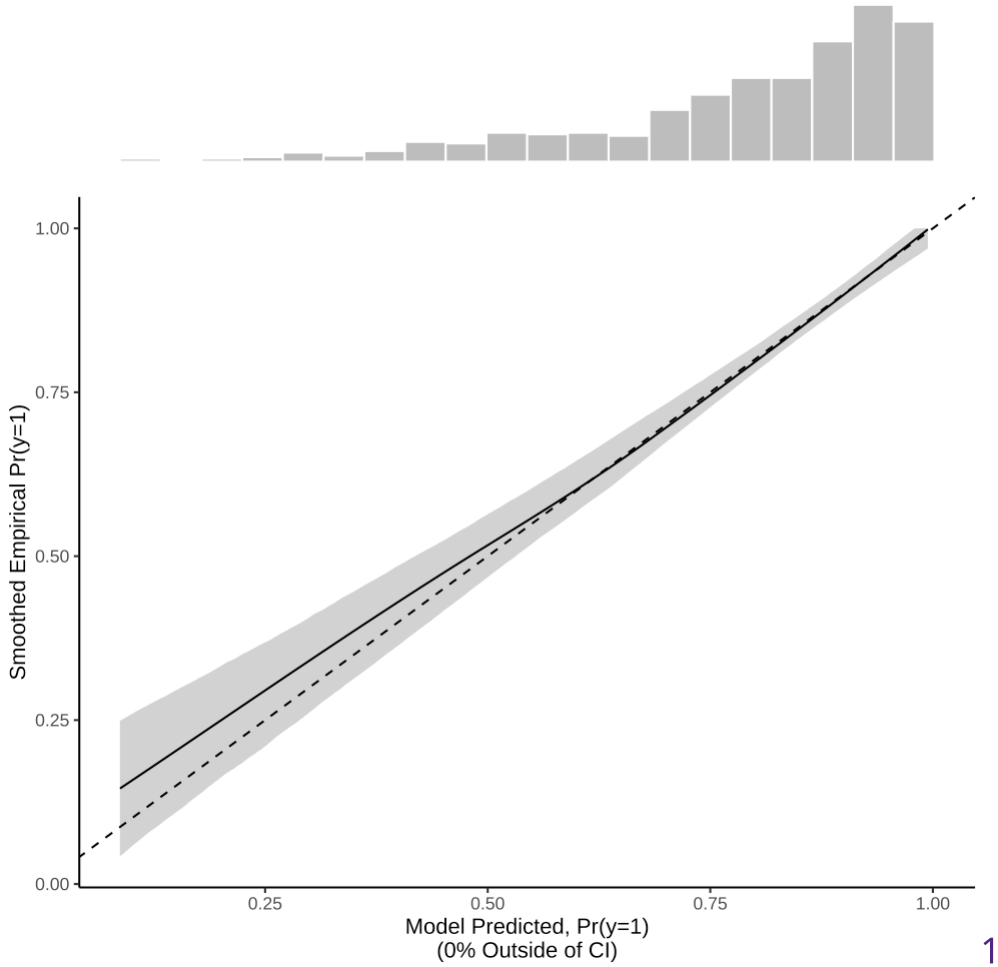
```
cols <- brewer.pal(5, "Set2")[c(2,3)]
y <- model.response(model.frame(mod))
separationplot(fitted(mod), c(y),
    col0=cols[1], col1=cols[2], file=NULL)
```



Empirical vs Predicted Probabilities

```
# remotes::install_github("davidaarmstrong/psre")
library(psre)
gh1 <- gg_hmf(model.response(model.frame(mod)), fitted(mod), meth
```

```
library(patchwork)
gh1[[1]] + gh1[[2]] + plot_layout(heights=c(2,8), ncol=1)
```



Comparing Two Models: Nested

- Nested: use likelihood ratio test.

```
library(lmtest)
m1 <-  glm(voted ~ age + educ + income + poly(leftright, 2) +
    female + race, data=dat, family=binomial(link="logit"))
m2 <-  glm(voted ~ educ + income + poly(leftright, 2),
           data=dat, family=binomial(link="logit"))
lrtest(m1, m2)
```

```
## Likelihood ratio test
##
## Model 1: voted ~ age + educ + income + poly(leftright, 2) + female + race
## Model 2: voted ~ educ + income + poly(leftright, 2)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -249.80
## 2   5 -277.46 -4 55.311  2.797e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing Non-nested Models: AIC

Let's imagine that all models we are comparing are trying to explain the same objective reality (call it f).

- f is a function of ξ (the infinitely large set of all possible explanatory variables) and is non-parametric.

Further, imagine that we are estimating some model $g(\mathbf{X}|\theta)$ that predicts reality using $\mathbf{X}$, a subset of the variables in ξ and θ a set of parameters relating $\mathbf{X}$ to y. Then ...

$$\begin{aligned} I(f, g) &= E_f[\log(f(x))] - E_f[\log(g(x|\theta))] \\ &= C - E_f[\log(g(x|\theta))] \end{aligned}$$

Is the amount of information we lose when trying to approximate f with $g(\mathbf{X}, \theta)$.

AIC

Akaike found that the $LL(g)$, the log-likelihood of the model was a biased estimator of $I(f, g)$, but the bias was on the order of K (the number of parameters in the model). Therefore:

$$\widehat{I(f, g)} = LL(\theta|\mathbf{X}, y) - K$$

To increase similarity to the already well-established likelihood ratio statistic, Akaike multiplied his measure by -2:

$$AIC = -2LL(\theta|\mathbf{X}, y) + 2K$$

Small Sample Correction

There is a small sample correction that should be used when n is small in the absolute terms or n is small relative to K (e.g., $\frac{n}{K} \leq 40$)

$$AIC_c = -2LL(\theta|\text{data}) + 2K + \frac{2K(K+1)}{n-K-1}$$

Comparing Non-nested Models: BIC

BIC is meant to approximate the Bayes Factor:

$$\frac{\Pr(D|M_1)}{\Pr(D|M_2)} = \frac{\int \Pr(\theta_1|M_1)\Pr(D|\theta_1, M_1)d\theta_1}{\int \Pr(\theta_2|M_2)\Pr(D|\theta_2, M_2)d\theta_2}$$

The approximation is defined as:

$$BIC = -2LL(\theta|\mathbf{X}, y) + K\log(n)$$

Δ values.

Δ_i should be calculated such that for each model i in the model set,

$$\Delta_i = IC_i - IC_{\min}$$

Where IC_i is the chosen information criterion for model i. This gives the *best* model $\Delta_i = 0$

- This captures the information loss due to using model g_i rather than the best model, g_{min}.
- The large Δ_i, the less likely model i is the best approximation of reality f.

Conventional cut-off values for Δ_i are:

- $\Delta_i \leq 2$ indicates substantial support,
- $4 \leq \Delta_i \leq 7$ indicates less support,
- $\Delta_i \geq 10$ indicates essentially no support.

AIC or BIC?

The question of whether to use AIC or BIC is often left to how much you want to penalize additional model parameters. In actuality, the question is one of performance in picking the best (lowest information loss) model.

- When there are *tapering effects*, AIC is better
- When reality is simple with a *few big effects* captured by the highest posterior probability models, then BIC is often better.

Comparing Non-nested Models: Clarke's Test

Clarke (2003) puts forth a distribution-free test that is really a "paired sign test". The statistic is calculated as:

$$d_i = \log(\mathcal{L}_{\beta,x_i}) - \log(\mathcal{L}_{\gamma,z_i}) + (p-q)\left(\frac{log(n)}{2n}\right)$$

$$B = \sum_{i=1}^{n} I_{0,+\infty}(d_i)$$

- The d_i are the difference in individual log-likelihoods for the two models
- The second equation above counts up the number of positive d_i values.
- We are testing to see whether B is significantly bigger than a random binomial variable that has a $p = .5$ and n the same as the number of rows in $\mathbf{X}$ and $\mathbf{Z}$.

Clarke Test in R

```
library(clarkeTest)
m1 <-  glm(voted ~ age + female + race, data=dat,
           family=binomial(link="logit"))
m2 <-  glm(voted ~ educ + income + poly(leftright, 2),
           data=dat, family=binomial(link="logit"))
IC_delta(m1, m2)
```

```
##     df D_AIC D_AICc D_BIC
## ..1  5 31.31  31.31 31.31
## ..2  5  0.00   0.00  0.00
```

```
clarke_test(m1, m2)
```

```
##
## Clarke test for non-nested models
##
## Model 1 log-likelihood: -293
## Model 2 log-likelihood: -277
## Observations: 584
## Test statistic: 246 (42%)
##
## Model 2 is preferred (p = 0.00016)
```

Comparing Non-nested Models: Cross-validation

```
library(rsample)
library(tidyr)
library(purrr)
dat$lrstren <- abs(dat$leftright-5)
m1 <-  glm(voted ~ age + female + race +
             lrstren, data=dat,
           family=binomial(link="logit"))
m2 <-  glm(voted ~ educ + income +
             poly(leftright, 2), data=dat,
           family=binomial(link="logit"))

cv_logit <- function(split, m1, m2, ...){
  m1_up <- update(m1, data=analysis(split))
  m2_up <- update(m2, data=analysis(split))
  y_out <- model.response(
            model.frame(formula(m1),
                        data=assessment(split)))
  p1 <- predict(m1_up,
                newdata=assessment(split),
                type="response")
  p2 <- predict(m2_up,
                newdata=assessment(split),
                type="response")
  ll1 <- sum(-y_out*log(p1) - (1-y_out)*log(1-p1))
  ll2 <- sum(-y_out*log(p2) - (1-y_out)*log(1-p2))
  tibble(model = factor(1:2,
                        labels=c("Model1", "Model2")),
         ll = c(ll1, ll2))
}
```

```
cv_out <- vfold_cv(dat,
                   v=10,
                   repeats=10) %>%
    mutate(ll = map(splits, cv_logit, m1, m2)) %>%
    unnest(ll) %>%
    group_by(id, model) %>%
    summarise(ll = sum(ll)) %>%
    pivot_wider(names_from="model", values_from="ll") %>%
  mutate(diff = Model2-Model1)
cv_out
```

```
## # A tibble: 10 × 4
## # Groups:   id [10]
##    id       Model1 Model2   diff
##    <chr>     <dbl>  <dbl>  <dbl>
##  1 Repeat01   293.   283.  -9.83
##  2 Repeat02   293.   283.  -9.91
##  3 Repeat03   296.   282. -14.4
##  4 Repeat04   294.   285.  -9.44
##  5 Repeat05   294.   284.  -9.92
##  6 Repeat06   295.   283. -12.4
##  7 Repeat07   294.   283. -11.8
##  8 Repeat08   294.   284. -10.3
##  9 Repeat09   294.   283. -11.2
## 10 Repeat10   297.   286. -11.3
```

Residuals, Outliers and Influential Observations

While looking at the residuals is marginally less interesting in these models, they can still tell us something about model fit.

- Many different kinds of residuals, all tell us something different about the model.

Response and Pearson Residuals

- Response residuals are: $y_i - \hat{p}_i$, but have little diagnostic value because they do not account for the inherent heteroskedasticity of the non-linear models.
- Pearson Residuals: $e_{Pi} = \frac{y_i - \hat{p}_i}{\text{var}(y_i|\mathbf{x})} = \frac{y_i - \hat{p}_i}{\sqrt{\hat{p}_i(1-\hat{p}_i)}}$
- Standardized Pearson Residuals: $e_{PSi} = \frac{e_{Pi}}{\sqrt{1-h_i}}$ where $h_i = \hat{p}_i(1-\hat{p}_i)x_i\widehat{\text{Var}}\left(\hat{\beta}\right)x_i'$

Deviance Residuals

The deviance residuals are the observations contribution to the overall deviance:

$$e_{Di} = \text{sign}(y - \hat{p}_i)\sqrt{(2)}\sqrt{y_i \log\left(\frac{y_i}{\hat{p}_i}\right) + (1 - y_i)\log\left(\frac{1 - y_i}{1 - \hat{p}_i}\right)}$$

A standardized version of the deviance residual is also available:

$$e_{DSi} = \frac{e_{Di}}{\sqrt{1 - h_i}}$$

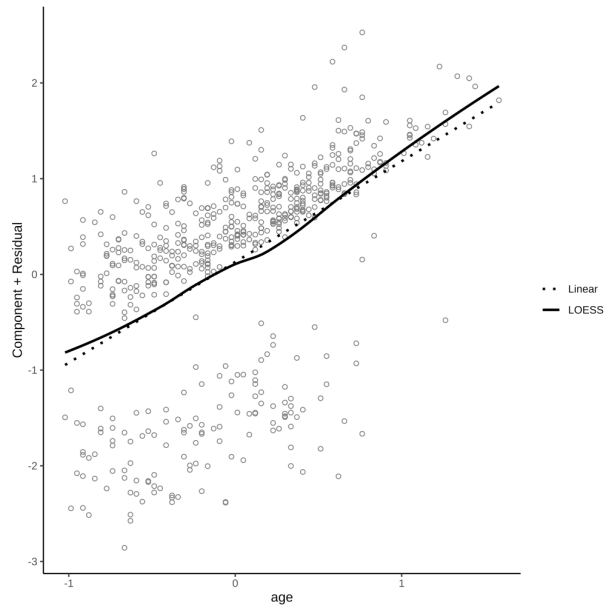
The standardized Pearson and deviance residuals can be obtained with `rstandard()` in R.

Component + Residual Plots

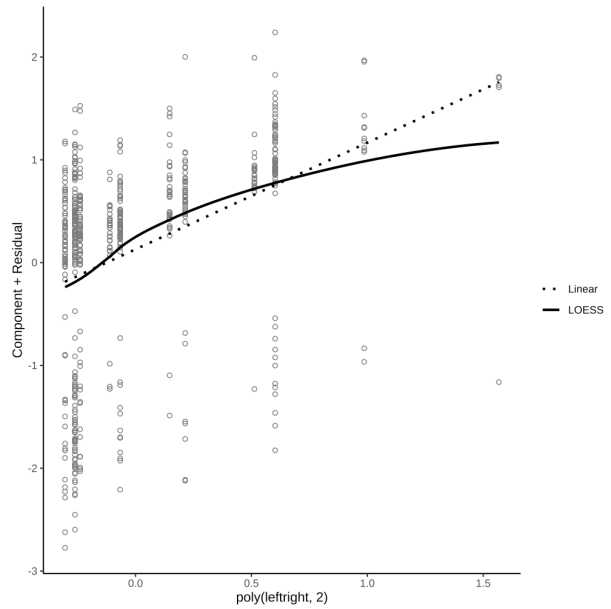
The `gg_crplot()` function is defined in the code for this class.

```
m1 <-  glm(voted ~ age + educ + income + poly(leftright, 2) +
    female + race, data=dat, family=binomial(link="logit"))
```

```
gg_crplot(m1, "age")
```




```
gg_crplot(m1, "poly(leftright, 2)")
```



New Model

```
dat$lrfac <- as.factor(dat$leftright)
mnew <- glm(voted ~ age + educ + income + as.factor(leftright) +
    female + race, data=dat, family=binomial(link="logit"))
anova(m1, mnew, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: voted ~ age + educ + income + poly(leftright, 2) + female + race
## Model 2: voted ~ age + educ + income + as.factor(leftright) + female +
##     race
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       575     499.61
## 2       567     483.31  8     16.3  0.03828 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Outlier Diagnostics

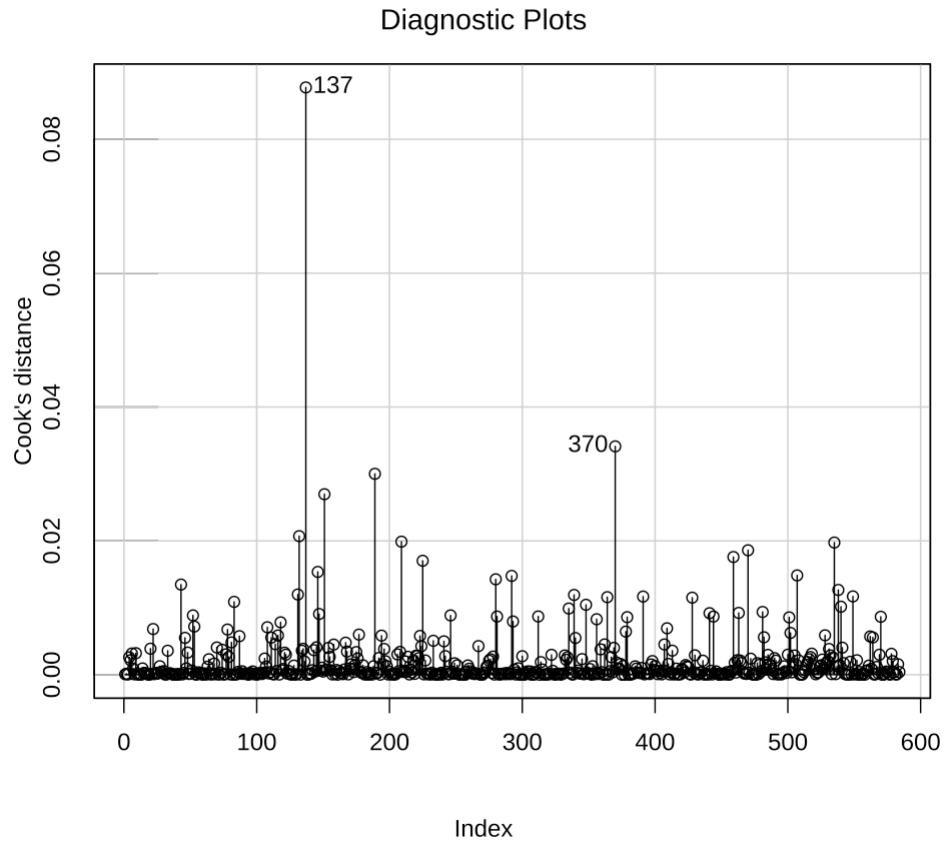
Cook's D can also be calculated for these models:

$$D_i = \frac{e^2_{PSi}}{K+1} \frac{h_i}{1-h_i}$$

The function `influenceIndexPlot` in R will produce an index plot of Cook's distances.

Residual Plots

```
library(car)
influenceIndexPlot(mod, vars="Cook", id.n=10)
```



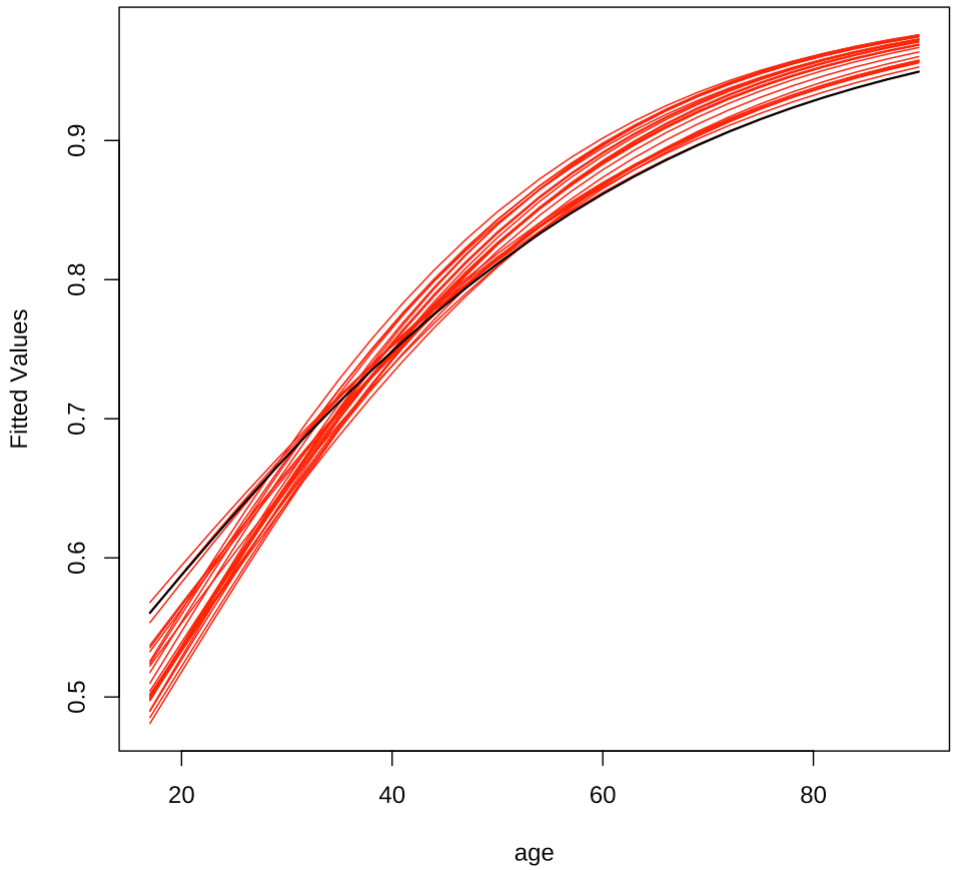
Looking at Deleted Obs Effects

I wrote a function called `outEff` that plots the effect of variables on predicted probabilities after removing observations that are thought to be most outlying.

- It can do this either one at a time or cumulatively.

outEff

```
outEff(mod,
       var='age',
       data=dat,
       nOut=25,
       cumulative=TRUE)
```



Separation/Small Sample Problems

MLE has a difficult time providing accurate (i.e., unbiased) estimates in the presence of small samples.

- Small samples in the traditional sense (overall small n).
- Small number of observations in the least populous category

Similar problems occur in a condition of separation (or quasi-separation).Zorn (2005) is a nice piece on this.

- No variability on the DV for some category of an independent variable.

Separation

The problem with separation is

- Parameter values try to take on values arbitrarily far away from zero (ultimately either $-\infty$ or ∞)

You know you have this problem if:

- You're using the `logit` command in Stata and it tells you that some observations were perfectly predicted and thus dropped.
- You're estimating a GLM in any software and you get huge coefficients with really huge standard errors.

Firth's Solution

Firth's solution is a penalized likelihood where:

- The penalty on the likelihood function drives parameters toward zero
- It does so at a much greater rate or those where separation exists (ultimately by providing relatively low weight to those cases).

The math is a bit beyond the scope of our discussion, but this is the accepted solution for separation problems across a wide range of disciplines.

Example: Voting for National Front

```
sepdat <- rio::import("data/france_binary.dta")
sepdat <- sepdat %>%
  mutate(across(c("demsat", "retnat", "union"), rio::factorize))
mod3 <- glm(votefn ~ demsat + age + lrself + hhincome + retnat
    + union, data=sepdat, family=binomial)
printCoefmat(summary(mod3)$coefficients)
```

```
##                               Estimate  Std. Error z value Pr(>|z|)
## (Intercept)                -19.7125946 689.5448696 -0.0286   0.9772
## demsatSomewhat satisfied    15.2255675 689.5443893  0.0221   0.9824
## demsatA little satisfied    16.0828401 689.5444023  0.0233   0.9814
## demsatNot at all satisfied  16.8229674 689.5444325  0.0244   0.9805
## age                         -0.0039201   0.0084709 -0.4628   0.6435
## lrself                       0.2537232   0.0515159  4.9251 8.43e-07 ***
## hhincome                    -0.1305836   0.0860719 -1.5171   0.1292
## retnatsame                   0.5459725   0.6085426  0.8972   0.3696
## retnatworse                  0.8501427   0.5649549  1.5048   0.1324
## unionYes                    -0.3255606   0.5036256 -0.6464   0.5180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Firth Logit

```
library(logistf)
mod3a <- logistf(mod3, sepdat)
pfl(mod3a)
```

```
##                              coef    se(coef) p      lower 0.95 upper 0.95
## (Intercept)                 -6.1860 1.5819   0.0000 -11.1977   -3.5833
## demsatSomewhat satisfied     1.8711 1.3854   0.0762  -0.1462    6.7233
## demsatA little satisfied     2.7207 1.3916   0.0036   0.6813    7.5774
## demsatNot at all satisfied   3.4529 1.4053   0.0001   1.3640    8.3195
## age                         -0.0037 0.0081   0.6573  -0.0203    0.0126
## lrself                       0.2483 0.0494   0.0000   0.1504    0.3499
## hhincome                    -0.1280 0.0820   0.1300  -0.2964    0.0375
## retnatsame                   0.4700 0.5603   0.4047  -0.6084    1.7154
## retnatworse                  0.7421 0.5196   0.1384  -0.2207    1.9295
## unionYes                    -0.2480 0.4669   0.5978  -1.3016    0.6217
##                             Chisq
## (Intercept)                 32.4146
## demsatSomewhat satisfied     3.1445
## demsatA little satisfied     8.4590
## demsatNot at all satisfied 14.9172
## age                          0.1968
## lrself                      25.2481
## hhincome                     2.2929
## retnatsame                   0.6944
## retnatworse                  2.1958
## unionYes                     0.2783
```

Review

1. Absolute Measures of Model Fit
2. Comparative Measures of Model Fit
3. Nested and Non-nested Model Tests
4. Model Specification Tests
5. Model Diagnostics
6. (Quasi-)Separation

Exercises

1. Estimate a model of `cwmid` on `lnwaterpcmin`, `instcoop`, `numbtreaties`, `anyupdown`, `power1`, `alliance`, `gdpmax`, `interdep`, `dyaddem`, `contig`, `peaceyrs1`, `_spline1`, `_spline2` and `_spline3` included additively.
    - Treat the `instcoop` variable both as continuous and categorical in different models.
    - How do these two models compare?
2. Add the interaction of `instcoop` and `lnwaterpcmin` to the two models from question 1.
    - How do these two models compare?
    - Of the four models, which is the best one?
3. Consider problems of non-linearity and outliers in the best model.
    - What do you find?