

Regression III: Lab 2

Dave Armstrong
University of Wisconsin – Milwaukee
Department of Political Science
e: armstrod@uwm.edu
w: www.quantoid.net/ICPSR.php

This lab exercise will help you wrap your head around three different things we've talked about 1) robust regression, 2) bootstrapping and cross-validation and 3) mixture models/model selection. You may want to start on the set of problems that most interests you and move to others as time permits. The data in the file are as follows:

country Country name

ccode Country numeric code from Correlates of War (COW)

year Year

rep1 State Repression (higher values indicate higher repression).

voice Democratic behavior (higher values indicate more democratic behavior).

veto Democratic institutions (higher values indicate more democratic institutions).

gdppc GDP/capita in constant USD.

pop Population (in thousands).

cwar Involvement in a civil war (COW).

iwar Involvement in an interstate war (COW).

1 Bootstrapping and Cross-Validation

Using the data from the last homework, I want you to investigate the following set of questions. You can read the data in with:

```
> library(foreign)
> boot.data <- read.dta("http://www.quantoid.net/boot_data.dta")
> rownames(boot.data) <- boot.data$country
```

1. Estimate the regression of `rep1` on `gdppc`, `logpop`, `iwar`, `cwar` and the interaction of `voice` and `veto`. Evaluate whether a multiplicative smooth function is “better” than a multiplicative linear specification.

```
> library(mgcv)
> library(boot)
> mod.lm <- gam(rep1 ~ gdppc + logpop + iwar + cwar + voice * veto,
+ data = boot.data)
> mod.gam <- gam(rep1 ~ gdppc + logpop + iwar + cwar + s(voice,
+ veto), data = boot.data)
> anova(mod.lm, mod.gam, test = "Chisq")1
```

Analysis of Deviance Table

```
Model 1: rep1 ~ gdppc + logpop + iwar + cwar + voice * veto
Model 2: rep1 ~ gdppc + logpop + iwar + cwar + s(voice, veto)
```

	Resid. Df	Resid. Dev	Df	Deviance	P(> Chi)
1	105.0	69.805			
2	103.5	65.008	1.4973	4.797	0.01212 *

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `anova` command suggests that there are significant differences between the two models. However, we may not want to make distributional assumptions, thus we may want to use bootstrapping to figure out whether the models are significantly different from each other.

```
> resid1 <- mod.lm$residuals
> yhat <- mod.lm$fitted
> test.stat <- deviance(mod.lm) - deviance(mod.gam)
> boot.gam <- function(dat, inds) {
+   assign(".inds", inds, envir = .GlobalEnv)
+   boot.e <- resid1[.inds]
+   boot.y <-<- yhat + boot.e
+   g1 <- update(mod.lm, boot.y ~ .)
+   g2 <- update(mod.gam, boot.y ~ .)
+   remove(".inds", envir = .GlobalEnv)
+   deviance(g1) - deviance(g2)
+ }
> boot.g <- boot(boot.data, boot.gam, R = 1000)
> (sum(boot.g$t >= test.stat) + 1)/(1000 + 1)
```

```
[1] 0.1248751
```

This shows us that there are actually not significant differences between the two models .

2. *Operating under the assumption that the GAM is better than the LM, how would you plot the interactive relationship between voice, veto and rep1 with 95% confidence intervals?*

```
> voice.seq <- seq(min(boot.data$voice), max(boot.data$voice),
+   length = 25)
> veto.seq <- seq(min(boot.data$veto), max(boot.data$veto), length = 25)
> pred.dat <- expand.grid(voice = voice.seq, veto = veto.seq, gdppc = mean(boot.data$gdppc,
+   na.rm = T), logpop = mean(boot.data$logpop, na.rm = T), iwar = 0,
+   cwar = 0)
> resid1 <- mod.gam$residuals
> yhat <- mod.gam$fitted
> boot.preds <- function(dat, inds) {
+   assign(".inds", inds, envir = .GlobalEnv)
+   boot.e <- resid1[.inds]
+   boot.y <-<- yhat + boot.e
+   g2 <- update(mod.gam, boot.y ~ .)
+   remove(".inds", envir = .GlobalEnv)
+   predict(g2, newdata = pred.dat)
+ }
> boot.p <- boot(boot.data, boot.preds, R = 100)
> boot.ci <- t(sapply(1:625, function(x) boot.ci(boot.p, index = x,
+   type = "perc")$perc[4:5])))
```

```

> fit <- predict(mod.gam, newdata = pred.dat)
> plot.data <- data.frame(fit = c(fit, boot.cis[, 1], boot.cis[,
+   2]), type = rep(c("fit", "lower", "upper"), each = 625),
+   voice = rep(pred.dat$voice, 3), veto = rep(pred.dat$veto,
+   3))
> library(lattice)
> cols <- c("black", "#FF000033", "#FF000033")
> wireframe(fit ~ voice + veto, groups = type, data = plot.data,
+   screen = list(x = -90, z = 0, y = -50), col = cols, col.groups = cols)

```

3. If, instead of using a multiplicative GAM, you wanted to model the relationship as an additive function of voice and veto with splines. Use cross-validation to figure out the appropriate number of knots.

```

> library(splines)
> res <- NULL
> kts <- expand.grid(1:10, 1:10)
> for (i in 1:nrow(kts)) {
+   {
+     if (kts[i, 1] < 4) {
+       tmp.voice <- with(boot.data, poly(voice, kts[i, 1]))
+     }
+     else {
+       tmp.voice <- with(boot.data, bs(voice, df = kts[i,
+       1]))
+     }
+   }
+   {
+     if (kts[i, 2] < 4) {
+       tmp.veto <- with(boot.data, poly(veto, kts[i, 2]))
+     }
+     else {
+       tmp.veto <- with(boot.data, bs(veto, df = kts[i,
+       2]))
+     }
+   }
+   colnames(tmp.veto) <- paste("ve", 1:ncol(tmp.veto), sep = "")
+   colnames(tmp.voice) <- paste("vo", 1:ncol(tmp.voice), sep = "")
+   tmp.data <- cbind(boot.data, tmp.veto, tmp.voice)
+   form <- paste("rep1 ~ gdppc + logpop + \n\tiwar + cwar +",
+     paste(c(colnames(tmp.voice), colnames(tmp.veto)), collapse = " + "),
+     sep = "")
+   tmp.mod <- mod1 <- gam(as.formula(form), data = tmp.data)
+   res <- rbind(res, cv.glm(tmp.data, tmp.mod, K = 5)$delta)
+ }
> kts[which.min(res[, 2]), ]

```

```

      Var1 Var2
22      2    3

```

Now that we know 5 degrees of freedom for voice and 2 df for veto is the right specification from a CV point of view, let's look at the model.

```

> cvmod <- lm(rep1 ~ gdppc + logpop + iwar + cwar + bs(voice, df = 5) +
+   poly(veto, 2), data = boot.data)
> summary(cvmod)

```

```
Call:
lm(formula = rep1 ~ gdppc + logpop + iwar + cwar + bs(voice,
  df = 5) + poly(veto, 2), data = boot.data)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-1.9651 -0.5483 -0.0611  0.5923  1.9079
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.408e+00  8.499e-01  -2.833  0.00557 **
gdppc          -7.299e-05  1.366e-05  -5.343  5.67e-07 ***
logpop         3.386e-01  5.580e-02   6.068  2.27e-08 ***
iwar           1.096e+00  4.113e-01   2.665  0.00898 **
cwar           9.277e-01  4.588e-01   2.022  0.04581 *
bs(voice, df = 5)1  1.748e-01  1.137e+00   0.154  0.87817
bs(voice, df = 5)2 -9.864e-01  6.300e-01  -1.566  0.12053
bs(voice, df = 5)3  1.922e-01  9.809e-01   0.196  0.84507
bs(voice, df = 5)4 -2.658e+00  8.039e-01  -3.306  0.00131 **
bs(voice, df = 5)5 -1.482e+00  9.897e-01  -1.498  0.13730
poly(veto, 2)1    -3.402e+00  2.090e+00  -1.628  0.10667
poly(veto, 2)2     1.168e+00  1.046e+00   1.116  0.26692
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.7908 on 101 degrees of freedom
Multiple R-squared:  0.8438,    Adjusted R-squared:  0.8267
F-statistic: 49.58 on 11 and 101 DF,  p-value: < 2.2e-16
```

We could also calculate the CV error for the linear model with the interaction to see whether this is better:

```
> cv.glm(boot.data, mod.lm, K = 5)$delta
      1      1
0.7430090 0.7281171

> res[which.min(res[, 2]), ]
      1      1
0.6688911 0.6557371
```

2 Robust Regression and Outliers

1. Using the same data and the fully parametric model (i.e., one you could run with `lm`) you found to be best in the last step of the previous problem above evaluate the extent to which there may be outliers in this model using the numerical and graphical methods we talked about in class. If you're starting here (i.e., you didn't do the previous question), use a model that is additive in voice and veto, but potentially nonlinear in both of those variables.

```
> outmod <- lm(rep1 ~ gdppc + logpop + iwar + cwar + bs(voice,
+   df = 5) + poly(veto, 2), data = boot.data)
> library(car)
> outlierTest(outmod)
```

No Studentized residuals with Bonferonni $p < 0.05$
 Largest |rstudent|:

	rstudent	unadjusted p-value	Bonferonni p
Israel	2.687451	0.008434	0.95304

This appears to suggest that there are no real outliers (if we look at the Bonferroni correction). We could look at our other various numerical summaries:

```
> hats <- hatvalues(outmod)
> hats[which(hats > 2 * (outmod$rank)/length(outmod$residuals))]
```

United States of America	Cuba	Guatemala
0.2142506	0.3939054	0.2913786
Paraguay	United Kingdom	Belgium
0.2603792	0.2899640	0.5480143
Spain	Liberia	Libya
0.2858859	0.3311114	0.3861491
Sudan	United Arab Emirates	China
0.2133976	0.2404288	0.2698785
Myanmar		
0.2203003		

```
> rstud <- rstudent(outmod)
> rstud[which(abs(rstud) > 2)]
```

Poland	Ethiopia	Madagascar	Iraq
-2.618303	-2.408379	-2.043618	2.158105
Israel	Papua New Guinea		
2.687451	2.408452		

```
> d <- cooks.distance(outmod)
> d[which(d > 4/outmod$df.residual)]
```

Cuba	Belgium	Italy	Liberia
0.05269157	0.07071613	0.05164739	0.09183299
Ethiopia	Libya	Iraq	Israel
0.05806145	0.04790369	0.05155275	0.09565744
Papua New Guinea			
0.05305005			

```
> dfb <- dfbeta(outmod)
> apout <- apply(dfb, 2, function(x) x[which(abs(x) > 2/sqrt(length(outmod$residuals)))]))
> apout
```

```
$`(Intercept)`
```

Cuba	Libya	Israel	Bahrain
-0.3619633	0.3590569	0.1971999	0.2279047
Oman	Myanmar	Papua New Guinea	
-0.2113792	-0.2630238	0.2055076	

```
$gdppc
named numeric(0)
```

```
$logpop
named numeric(0)
```

\$iwar
Liberia
0.3496861

\$cwar
Liberia Israel
-0.3510222 0.2114177

\$`bs(voice, df = 5)1`
Cuba Burkina Faso Ethiopia Libya Kuwait
0.7609972 -0.2469809 0.3378755 -0.7154214 0.3227507

\$`bs(voice, df = 5)2`
Ethiopia Bahrain Oman Myanmar
-0.2043661 -0.1956049 0.2308269 0.2608942

\$`bs(voice, df = 5)3`
Cuba Mexico Paraguay Ethiopia
0.4628240 -0.1926394 -0.2873928 0.1936518
Libya Myanmar Papua New Guinea
-0.4126765 0.3210280 -0.2133470

\$`bs(voice, df = 5)4`
Cuba Belgium Libya Myanmar
0.3078149 0.1881903 -0.3060215 0.2146177
Papua New Guinea
0.2397637

\$`bs(voice, df = 5)5`
Cuba Paraguay Belgium Italy Libya Myanmar
0.3951098 -0.2387222 -0.5903860 0.4117429 -0.3787019 0.2450163

\$`poly(veto, 2)1`
Mexico Honduras Venezuela Paraguay
0.4066609 -0.2639165 -0.2453975 0.9405046
Chile Argentina Poland Moldova
0.2682996 -0.1991125 -0.1954944 -0.3553335
Estonia Ukraine Mali Liberia
-0.2666631 0.2444569 -0.2332980 -0.6008824
Ghana Kenya Botswana Madagascar
0.4473702 -0.5668407 -0.2174827 0.2843053
Iraq Saudi Arabia Kazakhstan China
-0.4885019 0.3257227 0.4137809 -0.2116748
North Korea India Myanmar Thailand
-0.3222437 0.1923787 -0.5928544 -0.2064786
Singapore Papua New Guinea
-0.2080342 0.3097374

\$`poly(veto, 2)2`
Chile Poland Liberia Madagascar Iraq Saudi Arabia
0.2101510 -0.2693893 0.2403563 0.1903141 0.3940449 -0.1905004
Kuwait Oman Kazakhstan North Korea Myanmar
-0.1983663 0.1910422 -0.1973451 0.2664611 0.3778039

```
> tab <- table(unlist(lapply(apout, names)))
> tab <- tab[order(tab)]
```

This suggests that there are certainly some outliers. We could think about influence on the model as a whole as the number of coefficients on which observations have “big” DF Beta values. We could do this the following way.

```
> tab <- table(unlist(lapply(apout, names)))
> tab <- tab[order(tab)]
> tab[which(tab > 2)]
```

Ethiopia	Oman	Paraguay	Liberia
3	3	3	4
Papua New Guinea	Cuba	Libya	Myanmar
4	5	5	7

Next, we could make the influence plot:

```
> pdf("labinfl1.pdf", height = 6, width = 6)
> influencePlot(outmod, id.n = 5)
```

	StudRes	Hat	CookD
Cuba	-0.9862264	0.39390538	0.22954644
Guatemala	-0.2332529	0.29137861	0.04338112
Belgium	-0.8353444	0.54801431	0.26592505
Poland	-2.6183032	0.04697886	0.16315166
Liberia	1.5012704	0.33111137	0.30303959
Ethiopia	-2.4083791	0.11176658	0.24095943
Libya	0.9555248	0.38614913	0.21886912
Iraq	2.1581048	0.12098535	0.22705231
Israel	2.6874515	0.14436770	0.30928537
Papua New Guinea	2.4084517	0.10310924	0.23032595

```
> invisible(dev.off())
```

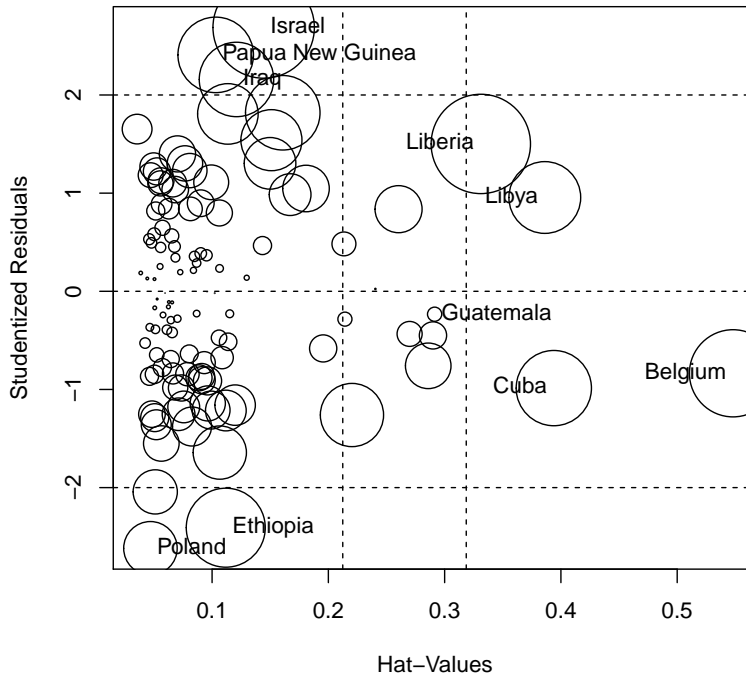
2. Now, run the robust regression to see if the same set of outliers are identified.

```
> library(MASS)
> robmod <- rlm(rep1 ~ gdppc + logpop + iwar + cwar + bs(voice,
+ df = 5) + poly(veto, 2), data = boot.data, method = "MM")
> pdf("weightplot.pdf", height = 6, width = 6)
> plot(robmod$w, ylab = "Observation Weight from MM-Estimation")
> ind <- which(robmod$w < 0.8)
> text((1:length(robmod$w))[ind], robmod$w[ind], names(robmod$fitted)[ind],
+ pos = 1)
> invisible(dev.off())
```

3. Now, since you have some non-linearity in either voice and/or veto, use either fixed-X bootstrapping to show how the effects of voice and veto change from the model you started this section with and the robust linear model. If this is the relationship we care about, are the outliers changing what we think is happening here?

```
> ols.e1 <- residuals(outmod)
> ols.fit1 <- fitted(outmod)
> rlm.e1 <- residuals(robmod)
> rlm.fit1 <- fitted(robmod)
```

Figure 1: Influence Plot

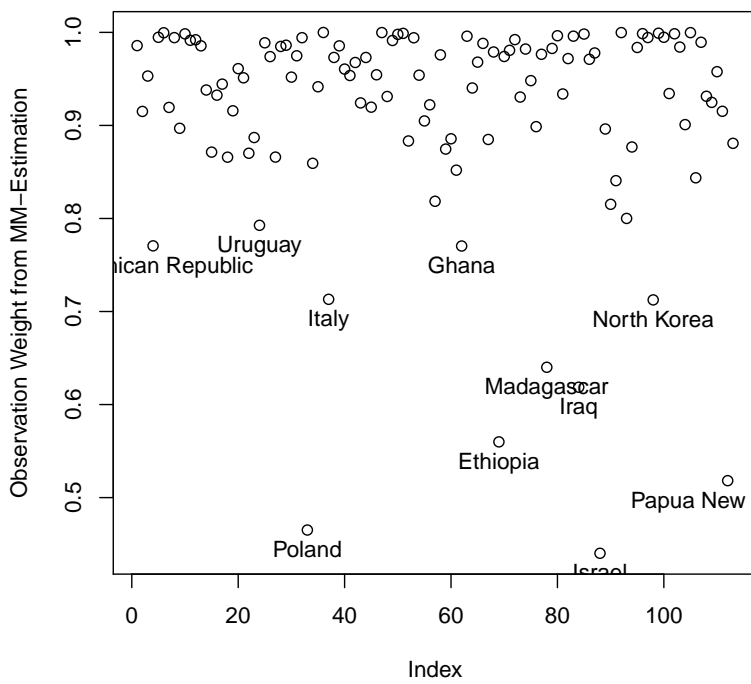


```

> pred.dat <- data.frame(voice = c(voice.seq, rep(mean(boot.data$voice),
+   25)), veto = c(rep(mean(boot.data$veto), 25), veto.seq),
+   gdppc = mean(boot.data$gdppc), logpop = mean(boot.data$logpop),
+   cwar = 0, iwar = 0)
> fix.boot.ols <- function(data, inds) {
+   assign(".inds", inds, envir = .GlobalEnv)
+   boot.e <- ols.e1[.inds]
+   boot.y <-< ols.fit1 + boot.e
+   tmp.mod <- update(outmod, boot.y ~ .)
+   remove(".inds", envir = .GlobalEnv)
+   predict(tmp.mod, newdata = pred.dat)
+ }
> fixx.ols1 <- boot(boot.data, fix.boot.ols, R = 1000)
> fix.boot.rlm <- function(data, inds) {
+   assign(".inds", inds, envir = .GlobalEnv)
+   boot.e <- rlm.e1[.inds]
+   boot.y <-< rlm.fit1 + boot.e
+   tmp.mod <- update(robmod, boot.y ~ ., maxit = 250)
+   remove(".inds", envir = .GlobalEnv)
+   predict(tmp.mod, newdata = pred.dat)
+ }
> fixx.rlm1 <- boot(boot.data, fix.boot.rlm, R = 1000)
> ols.ci <- t(sapply(1:50, function(x) boot.ci(fixx.ols1, index = x,
+   type = "perc")$perc[4:5]))
> rlm.ci <- t(sapply(1:50, function(x) boot.ci(fixx.rlm1, index = x,

```

Figure 2: Plot of Weights from Robust Model



```

+   type = "perc")$perc[4:5]))
> ols.fit <- predict(outmod, newdata = pred.dat)
> rlm.fit <- predict(robmod, newdata = pred.dat)
> plot.dat <- data.frame(fit = c(ols.fit, rlm.fit), lower = c(ols.ci[,
+   1], rlm.ci[, 1]), upper = c(ols.ci[, 2], rlm.ci[, 2]), vals = rep(c(voice.seq,
+   veto.seq), 2), var = rep(rep(c("voice", "veto"), each = 25),
+   2), mod = rep(c("OLS", "MM"), each = 50))
> library(lattice)
> library(latticeExtra)
> pdf("robplot.pdf", height = 6, width = 6)
> trellis.par.set(strip.background = list(col = "White"))
> print(useOuterStrips(xyplot(fit ~ vals | mod * var, data = plot.dat,
+   scales = list(x = "free"), ylim = range(c(plot.dat[, c("lower",
+   "upper")])) * 1.1, panel = function(x, y, subscripts) {
+   panel.lines(x, y, col = "black")
+   panel.lines(x, plot.dat$lower[subscripts], col = "black",
+   lty = 2)
+   panel.lines(x, plot.dat$upper[subscripts], col = "black",
+   lty = 2)
+   })))
> invisible(dev.off())

```

3 Mixture Models

1. Now, continuing with these data, estimate a finite mixture model where voice and veto are treated as competing, rather than complementary theories. Force the coefficients for the other variables (*gdppc*, *logpop*, *cwar* and *iwar*) to be constant across the two latent groups. Evaluate the fit of the model and the evidence for each theory. To use the *IdentifyList* command, do the following:

```

> source("http://www.quantoid.net/mixtureTools.R")

> library(flexmix)
> boot.data$gdppc <- boot.data$gdppc/10000
> model <- FLXMRglmfix(family = "gaussian", nested = list(k = c(1,
+   1), formula = c(~voice, ~veto)), fixed = ~gdppc + logpop +
+   iwar + cwar)
> out <- stepFlexmix(rep1 ~ 1, k = 2, model = model, data = boot.data,
+   nrep = 20)

```

```
2 : * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
> summary(out)
```

Call:

```
stepFlexmix(rep1 ~ 1, model = model, data = boot.data, k = 2,
  nrep = 20)
```

	prior	size	post>0	ratio
Comp.1	0.834	98	113	0.867
Comp.2	0.166	15	70	0.214

```
'log Lik.' -130.4952 (df=11)
AIC: 282.9903   BIC: 312.9916
```

We see many more observations are in the “voice” theory (consistent with the example yesterday).

```
> out.refit <- refit(out)
> summary(out.refit)
```

```
$Comp.1
```

	Estimate	Std. Error	z value	Pr(> z)	
gdppc	-0.910544	0.123941	-7.3466	2.033e-13	***
logpop	0.310542	0.038140	8.1422	3.882e-16	***
iwar	1.429161	0.305323	4.6808	2.857e-06	***
cwar	0.506364	0.321288	1.5760	0.115	
voice	-0.453492	0.054628	-8.3014	< 2.2e-16	***
(Intercept)	-3.199604	0.430911	-7.4252	1.126e-13	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
$Comp.2
```

	Estimate	Std. Error	z value	Pr(> z)	
gdppc	-0.91054	0.12394	-7.3466	2.033e-13	***
logpop	0.31054	0.03814	8.1422	3.882e-16	***
iwar	1.42916	0.30532	4.6808	2.857e-06	***
cwar	0.50636	0.32129	1.5760	0.1150	
veto	-0.27918	0.38048	-0.7338	0.4631	
(Intercept)	-1.72199	0.41417	-4.1577	3.215e-05	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Veto is not a significant predictor among its set of observations, but voice is a significant predictor among its set.

```
> idl <- IdentifyList(out@posterior$scaled, boot.data, case = boot.data$ccode,
+   cluster = F, alpha = 0.05)
```

You can see that there are a number of cases that are better predicted by the “voice” theory, but none that are better predicted by the “veto” theory.