PS04 by Emily Han

2025-01-31

1. Fearon & Laitin

You will need to download the file flmdw.csv and load it into your workspace. This is the data needed to replicate the analysis in Fearon & Laitin's 2003 APSR paper. The data are country-year data, but we will be ignoring the time component for the purposes of this exercise.

You will analyze the Fearon & Laitin data on civil war onset, specifically, the binary onset variable.

```
#Loading the dataset
dat <- read.csv("./data/flmdw.csv")
```

a. Examine the distribution of onset. Is this a "rare event"? What options might you consider?

```
table(dat$onset)
```

```
##
##    0    1
## 6296  106
```

b. Fit at least four models that predict binary onset. Analyze only complete cases and make sure that all three models are fit to the same observations but be careful how you remove NAs

Model 1 should be a logistic regression including only an intercept, GDP per capita (gdpenl), population (lpopl1) and percent mountainous (lmtnest).

```
logit_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest, family = binomial (link = 'logit'), data = dat)
```

Model 2 should be a probit regression but otherwise identical to model 1.

```
probit_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest, family = binomial (link = 'probit'), data = dat)
```

Model 3 should be a probit regression that adds a dummy variable for whether a country is an oil exporter (Oil), democracy (polity2l), and religious fractionalization (relfrac).

```
probit_dummy_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest + Oil + polity2l + relfrac, family = binomial
```

Model 4 should be a probit regression that includes an interaction between polity2l and relfrac.

```
probit_inter_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest + Oil + polity2l + relfrac + (relfrac * polit
```

```
#Models' summary
library(huxtable)

huxreg(list("Logit Model" = logit_mod, "Probit Model" = probit_mod, "Probit w/ Dummy" = probit_dummy_mo
```

c. Develop ROC plot and separation plots comparing the in-sample fit of your estimated models. The ROCR library may help

```
library(pROC)
```

	Logit Model	Probit Model	Probit w/ Dummy	Probit w/ Interaction
(Intercept)	-6.042 ***	-2.966 ***	-2.950 ***	-3.038 ***
	(0.612)	(0.256)	(0.274)	(0.280)
gdpenl	-0.295 ***	-0.114 ***	-0.140 ***	-0.147 ***
	(0.062)	(0.023)	(0.027)	(0.027)
lpopl1	0.236 ***	0.100 ***	0.088 **	0.094 ***
	(0.062)	(0.027)	(0.027)	(0.027)
lmtnest	0.178 *	0.075 *	0.085 *	0.090 **
	(0.080)	(0.032)	(0.033)	(0.033)
Oil			0.404 ***	0.403 ***
			(0.117)	(0.117)
polity2l			0.013 *	-0.006
			(0.007)	(0.013)
relfrac			0.222	0.318
			(0.199)	(0.202)
polity2l:relfrac				0.054
				(0.029)
N	6402	6402	6402	6402
logLik	-506.525	-506.218	-499.506	-497.828
AIC	1021.051	1020.436	1013.011	1011.656

*** p < 0.001; ** p < 0.01; * p < 0.05.

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

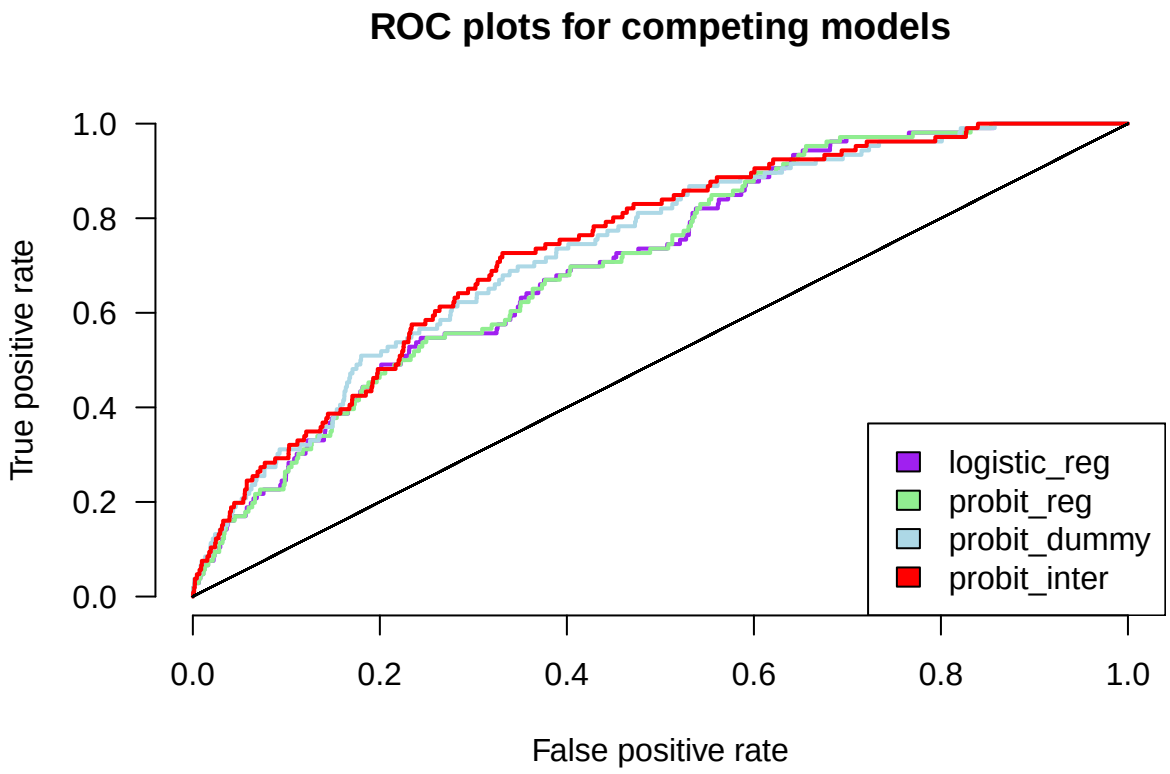
```
library(ROCR)
```

```
predicted_logit <-fitted(logit_mod)
predicted_probit <- fitted(probit_mod)
predicted_probit_dummy <- fitted(probit_dummy_mod)
predicted_probit_inter <- fitted(probit_inter_mod)
```

```r
actual <- as.vector(logit_mod$model$onset)
```

```r
pred_logit <- prediction(predicted_logit,actual)
perf_logit <- performance(pred_logit,"tpr","fpr")

pred_probit <- prediction(predicted_probit,actual)
perf_probit <- performance(pred_probit,"tpr","fpr")

pred_probit_dummy <- prediction(predicted_probit_dummy,actual)
perf_probit_dummy <- performance(pred_probit_dummy,"tpr","fpr")

pred_probit_inter <- prediction(predicted_probit_inter,actual)
perf_probit_inter <- performance(pred_probit_inter,"tpr","fpr")
```

```r
# Plotting the ROC
par(las=1, bty="n")
plot(perf_logit, main="ROC plots for competing models", bty="n",lwd=2, col = 'purple')
plot(perf_probit, lwd=2, col= 'lightgreen', add=T)
plot(perf_probit_dummy, lwd=2, col= 'lightblue', add=T)
plot(perf_probit_inter, lwd=2, col= 'red', add=T)
lines(actual,actual, lty="dashed")

legend("bottomright", legend = c("logistic_reg", "probit_reg", "probit_dummy", "probit_inter"), fill = c
```




```r
# Seperation Plot
library(DescTools)
```

```r
library(separationplot)
```

```
## Loading required package: RColorBrewer
```

```
## Loading required package: Hmisc
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:DescTools':
##
##     %nin%, Label, Mean, Quantile
```

```
## The following objects are masked from 'package:huxtable':
##
##     contents, label, label<-
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: MASS
```

```
## Loading required package: foreign
```

```r
#
# pred_logit_v <- pred_logit@predictions[[1]]
# pred_probit_v <-pred_probit@predictions[[1]]
# pred_probit_dummy_v <- pred_probit_dummy@predictions[[1]]
# pred_probit_inter_v <- pred_probit_inter@predictions[[1]]
#
#
#
#
#
# par(mfrow=c(4,2))
# separationplot(pred_logit_v,actual,
#                heading = "Party-only Model",
#                height = .50, width=2,
#                lwd1 = 0.9, newplot=F)
# separationplot(pred_probit_v,actual,
#                heading = "Full Model",
#                height = .50, width=2,
#                lwd1 =1, lwd2=1, newplot=F)
# separationplot(pred_probit_dummy_v,actual,
#                heading = "Full Model",
#                height = .50, width=2,
#                lwd1 =1, lwd2=1, newplot=F)
# separationplot(pred_probit_inter_v,actual,
#                heading = "Full Model",
#                height = .50, width=2,
#                lwd1 =1, lwd2=1, newplot=F)
```

d. Using model 4 and a likelihood ratio test, what is the evidence that we can leave polity2l out of the model entirely? In other words, test the hypothesis that

.

```
mod5 <- glm(onset ~ gdpenl + lpopl1 + lmtnest + Oil + relfrac, data = dat,
            family = "binomial"(link = "probit"))

ll_ratio <- (logLik(probit_inter_mod) - logLik(mod5))
ll_ratio_stat <- -2 * (ll_ratio)

ll_ratio_stat
```

```
## 'log Lik.' -7.355842 (df=8)
```

e.

Undertake a 10-fold cross-validation of each of these models. Construct an ROC plot of the out of sample predictive performance of each of the models. To do this you can write code to create the 10 folds or you can try and work with the tools in many R libraries that implement cross-validation. These include: cvTools, caret, tidymodels/resampling. On the basis of this analysis, which model(s) do you prefer?

```
probit_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest, family = binomial (link = 'probit'), data = dat)
```

Model 3 should be a probit regression that adds a dummy variable for whether a country is an oil exporter (Oil), democracy (polity2l), and religious fractionalization (relfrac).

```
probit_dummy_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest + Oil + polity2l + relfrac, family = binomial
```

Model 4 should be a probit regression that includes an interaction between polity2l and relfrac.

```
probit_inter_mod <- glm(onset ~ gdpenl + lpopl1 + lmtnest + Oil + polity2l + relfrac + (relfrac * polit
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:huxtable':
##
##     theme_grey
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
```

```
dat1 <-read.csv("./data/flmdw.csv")
dat1$onset <- as.factor(dat1$onset)
levels(dat1$onset) <- c("no", "yes")
```

```
# Define 10-fold cross-validation
cv_control <- trainControl(method = "cv", number = 10,
                           classProbs = TRUE, summaryFunction = twoClassSummary,
                           savePredictions = "final")
```

```
# Train Logit model with 10-fold CV
cv_logit <- train(onset ~ gdpenl + lpopl1 + lmtnest, data = dat1,
                  method = "glm", family = binomial(link = "logit"),
```

```
                  trControl = cv_control, metric = "ROC")

# Train Probit model with 10-fold CV
cv_probit <- train(onset ~ gdpenl + lpopl1 + lmtnest, data = dat1,
                   method = "glm", family = binomial(link = "probit"),
                   trControl = cv_control, metric = "ROC")

# Train Probit model with 10-fold CV
cv_probit_dummy <- train(onset ~ gdpenl + lpopl1 + lmtnest + Oil + polity2l + relfrac,
                         data = dat1, method = "glm", family = binomial(link = "probit"),
                         trControl = cv_control, metric = "ROC")

# Train Probit model with 10-fold CV
cv_probit_inter <- train(onset ~ gdpenl + lpopl1 + lmtnest + Oil + polity2l + relfrac + (relfrac * poli
                         data = dat1, method = "glm", family = binomial(link = "probit"),
                         trControl = cv_control, metric = "ROC")

# Extract Out-of-Sample ROC Data
roc_logit <- roc(cv_logit$pred$obs, cv_logit$pred$yes)   # Logit model OOS predictions
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
roc_probit <- roc(cv_probit$pred$obs, cv_probit$pred$yes) # Probit model OOS predictions
```

```
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
```

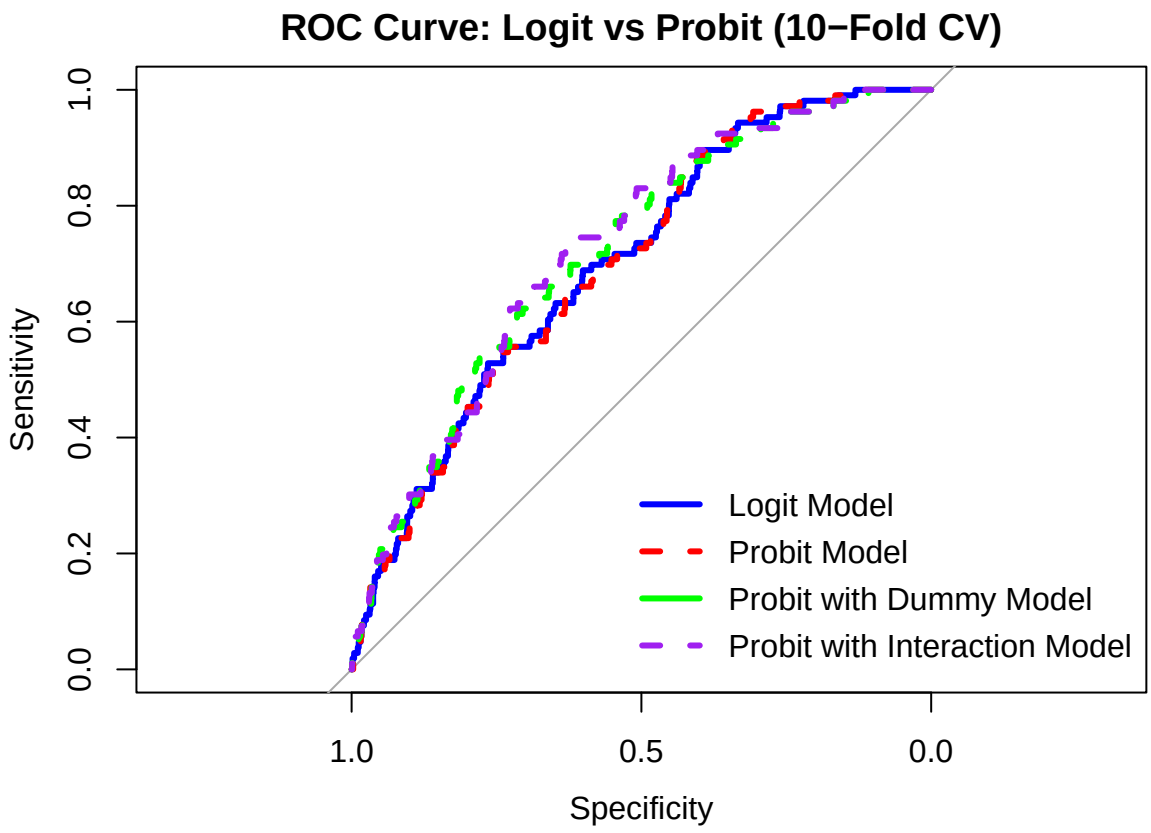
```
roc_probit_dummy <- roc(cv_probit_dummy$pred$obs, cv_probit_dummy$pred$yes)
```

```
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
```

```
roc_probit_inter <- roc(cv_probit_inter$pred$obs, cv_probit_inter$pred$yes)
```

```
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
```

```
# Plot ROC Curves for Logit vs Probit (10-Fold CV)
plot(roc_logit, col = "blue", lwd = 3, main = "ROC Curve: Logit vs Probit (10-Fold CV)")
plot(roc_probit, col = "red", lwd = 3, add = TRUE, lty = 2)
plot(roc_probit_dummy, col = "green", lwd = 3, add = TRUE, lty = 2)
plot(roc_probit_inter, col = "purple", lwd = 3, add = TRUE, lty = 2)


# Add legend
legend("bottomright", legend = c("Logit Model", "Probit Model", "Probit with Dummy Model", "Probit with
       col = c("blue", "red", "green", "purple"), lty = c(1, 2), lwd = 3, bty = "n")
```




```
# Print AUC values for model comparison
logit_auc <- auc(roc_logit)
probit_auc <- auc(roc_probit)
probit_dummy_auc <- auc(roc_probit_dummy)
probit_inter_auc <- auc(roc_probit_inter)

print(paste("Logit AUC:", round(logit_auc, 3)))
```

```
## [1] "Logit AUC: 0.701"
```

```
print(paste("Probit AUC:", round(probit_auc, 3)))
```

```
## [1] "Probit AUC: 0.701"
```

```
print(paste("Probit with Dummy AUC:", round(probit_dummy_auc, 3)))
```

```
## [1] "Probit with Dummy AUC: 0.718"
```

```
print(paste("Probit with Interaction AUC:", round(probit_inter_auc, 3)))
```

```
## [1] "Probit with Interaction AUC: 0.725"
```

f.

Interpret the relationship between civil war onset and democracy in the preferred model. Use a visual presentation of the model predictions and be sure to display the estimation uncertainty around the expected values produced by the model. Be sure to clearly state the scenarios you chose and why. If the best performing model includes the interaction term then provide a plot that interprets that conditional relationship. ## Use of ChatGPT and other generative AI tools

I certify that we did not use any LLM or generative AI tool in this assignment!

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: x86_64-apple-darwin20
## Running under: macOS 15.2
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] caret_7.0-1        lattice_0.22-6     ggplot2_3.5.1      separationplot_1.4
##  [5] foreign_0.8-86     MASS_7.3-60.2      Hmisc_5.1-3        RColorBrewer_1.1-3
##  [9] DescTools_0.99.58  ROCR_1.0-11        pROC_1.18.5        huxtable_5.5.7
##
## loaded via a namespace (and not attached):
##  [1] gridExtra_2.3        gld_2.6.7            readxl_1.4.3
##  [4] rlang_1.1.4          magrittr_2.0.3       furrr_0.3.1
##  [7] e1071_1.7-16         compiler_4.4.1       vctrs_0.6.5
## [10] reshape2_1.4.4       stringr_1.5.1        pkgconfig_2.0.3
## [13] crayon_1.5.3         fastmap_1.2.0        backports_1.5.0
## [16] utf8_1.2.4           rmarkdown_2.28       prodlim_2024.06.25
## [19] haven_2.5.4          purrr_1.0.2          xfun_0.48
## [22] recipes_1.1.0        highr_0.11           broom_1.0.7
## [25] parallel_4.4.1       cluster_2.1.6        R6_2.5.1
## [28] stringi_1.8.4        parallelly_1.38.0    boot_1.3-30
## [31] rpart_4.1.23         lubridate_1.9.3      cellranger_1.1.0
## [34] Rcpp_1.0.13          assertthat_0.2.1     iterators_1.0.14
## [37] knitr_1.48           future.apply_1.11.3  base64enc_0.1-3
## [40] Matrix_1.7-0         splines_4.4.1        nnet_7.3-19
## [43] timechange_0.3.0     tidyselect_1.2.1     rstudioapi_0.16.0
## [46] yaml_2.3.10          timeDate_4041.110    codetools_0.2-20
## [49] listenv_0.9.1        tibble_3.2.1         plyr_1.8.9
## [52] withr_3.0.1          evaluate_1.0.0       future_1.34.0
## [55] survival_3.6-4       proxy_0.4-27         pillar_1.9.0
## [58] stats4_4.4.1         checkmate_2.3.2      foreach_1.5.2
## [61] generics_0.1.3       hms_1.1.3            munsell_0.5.1
## [64] commonmark_1.9.2     scales_1.3.0         rootSolve_1.8.2.4
## [67] globals_0.16.3       class_7.3-22         glue_1.8.0
## [70] lmom_3.2             tools_4.4.1          data.table_1.16.2
## [73] ModelMetrics_1.2.2.2 gower_1.0.2          forcats_1.0.0
## [76] Exact_3.3            mvtnorm_1.3-1        grid_4.4.1
## [79] tidyr_1.3.1          ipred_0.9-15         colorspace_2.1-1
## [82] nlme_3.1-164         htmlTable_2.4.3      Formula_1.2-5
```

```
## [85] cli_3.6.3            fansi_1.0.6          expm_1.0-0
## [88] lava_1.8.1           dplyr_1.1.4          gtable_0.3.5
## [91] broom.mixed_0.2.9.5  digest_0.6.37        htmlwidgets_1.6.4
## [94] htmltools_0.5.8.1    lifecycle_1.0.4      hardhat_1.4.0
## [97] httr_1.4.7
```