Draper HW 4

Matthew Draper

April 29, 2019

```r
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 3.5.2
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.5.3
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.5.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(separationplot)
```

```
## Warning: package 'separationplot' was built under R version 3.5.3
```

```r
library(cvTools)
```

```
## Warning: package 'cvTools' was built under R version 3.5.3
```

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

```
## Loading required package: robustbase
```

```
## Warning: package 'robustbase' was built under R version 3.5.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

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

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```r
ms<-read.table("Msrepl87.asc", header=TRUE,
    colClasses=c("character",rep("numeric",22)))
rownames(ms) <- ms$country
```

```r
ms$sanctions <- (ms$sanctions70 + ms$sanctions75)/2
ms$deaths <- as.numeric(ms$deaths75 == 0)
```

```r
ms <- ms[complete.cases(ms), ]

m1 <- glm(deaths ~ sanctions, data = ms)

m2 <- glm(deaths ~ sanctions + giniland, data = ms)

m3 <- glm(deaths ~ sanctions + giniland + sanctions:giniland, data = ms)
```

```r
stargazer(m1, type = "html")
```

	Dependent variable:
	deaths
sanctions	-0.001^{*}
	(0.001)
Constant	0.266^{***}
	(0.069)
Observations	46
Log Likelihood	-22.006
Akaike Inf. Crit.	48.012
Note:	*p<0.1; **p<0.05;** p<0.01

```r
stargazer(m2, type = "html")
```

	Dependent variable:
	deaths
sanctions	-0.001^{*}
	(0.001)
giniland	-0.141
	(0.311)
Constant	0.351^{*}
	(0.201)
Observations	46
Log Likelihood	-21.896
Akaike Inf. Crit.	49.792
Note:	*p<0.1; **p<0.05;** p<0.01

```r
stargazer(m3, type = "html")
```

	Dependent variable:
	deaths
sanctions	-0.007
	(0.005)
giniland	-0.352
	(0.363)
sanctions:giniland	0.008
	(0.007)
Constant	0.504^{**}
	(0.242)
Observations	46
Log Likelihood	-21.213
Akaike Inf. Crit.	50.426
Note:	*p<0.1; **p<0.05;** p<0.01

We obtain very similar results for all three models. The AIC is lowest for m1, which is the simplest model (sanctions only). Including the Gini variable (m2) used up degrees of freedom without giving us anything useful (constant is not significant at p=.05). When we include the interaction term (m3), the constant is significant but the AIC is higher than in the simplest model.

```r
## Predicted Values
m1$fitted.values
```

```
##  United States         Canada        Jamaica         Mexico    El Salvador 
##    0.009887537    0.231814882    0.261577732    0.246696307    0.260930714 
##     Costa Rica         Panama       Colombia      Venezuela           Peru 
##    0.262871769    0.252519473    0.236344011    0.257695621    0.235049974 
##         Brazil      Argentina        Uruguay United Kingdom        Ireland 
##    0.214345382    0.211757308    0.236991030   -0.253448991    0.199463957 
##    Netherlands        Belgium         France    Switzerland          Spain 
##    0.242814196    0.260930714    0.138644218    0.254460529   -0.181629937 
##       Portugal   West Germany          Italy     Yugoslavia        Finland 
##    0.038356351    0.130879996    0.163877939    0.213698364    0.263518788 
##         Sweden         Norway        Denmark   Sierre Leone          Ghana 
##    0.257048603    0.260930714    0.261577732    0.258989658    0.246049289 
##          Kenya         Zambia         Malawi   South Africa         Turkey 
##    0.251872455    0.249284381    0.260930714    0.167760050    0.207228179 
##          Egypt    South Korea          Japan          India       Pakistan 
##    0.202699049    0.176818309    0.230520845    0.094646960    0.075236406 
##       Thailand       Malaysia    Philippines      Indonesia      Australia 
##    0.229226808    0.242167178    0.181347439    0.238285067    0.253813510 
##    New Zealand 
##    0.263518788
```

```r
length(m1$fitted.values)
```

```
## [1] 46
```

```r
m2$fitted.values
```

```
##  United States         Canada        Jamaica         Mexico    El Salvador 
##    0.002695061    0.246372708    0.234101575    0.201384426    0.232064818 
##     Costa Rica         Panama       Colombia      Venezuela           Peru 
##    0.232523785    0.228192195    0.201291314    0.214816904    0.192979322 
##         Brazil      Argentina        Uruguay United Kingdom        Ireland 
##    0.182903315    0.176169113    0.207566549   -0.247006339    0.218001815 
##    Netherlands        Belgium         France    Switzerland          Spain 
##    0.262630838    0.263146991    0.153701006    0.268210290   -0.193291095 
##       Portugal   West Germany          Italy     Yugoslavia        Finland 
##    0.017432587    0.149039487    0.146952139    0.221838513    0.313678799 
##         Sweden         Norway        Denmark   Sierre Leone          Ghana 
##    0.310265143    0.304118946    0.286376138    0.283880415    0.254447884 
##          Kenya         Zambia         Malawi   South Africa         Turkey 
##    0.243109350    0.227898192    0.298467642    0.157759854    0.211360726 
##          Egypt    South Korea          Japan          India       Pakistan 
##    0.195690601    0.221595102    0.257840281    0.098558270    0.095381429 
##       Thailand       Malaysia    Philippines      Indonesia      Australia 
##    0.250941115    0.262006907    0.197706098    0.245547887    0.223788752 
##    New Zealand 
##    0.245863150
```

```r
length(m2$fitted.values)
```

```
## [1] 46
```

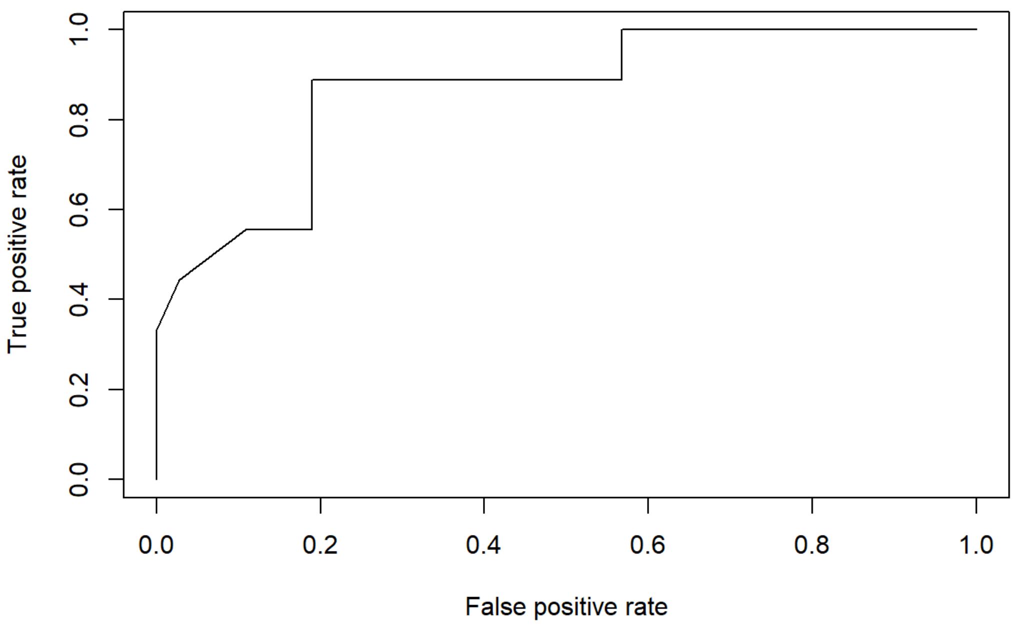
```r
m3$fitted.values
```

```
##  United States         Canada        Jamaica         Mexico    El Salvador
##     0.01425568     0.24900175     0.22051107     0.18367490     0.21702242
##     Costa Rica         Panama       Colombia      Venezuela           Peru
##     0.21441776     0.22197338     0.19935131     0.18571465     0.19114022
##         Brazil      Argentina        Uruguay United Kingdom        Ireland
##     0.19861015     0.19752147     0.20636942    -0.31350358     0.17648779
##    Netherlands        Belgium         France    Switzerland          Spain
##     0.28168135     0.28760468     0.05188515     0.29898892     0.02845797
##       Portugal   West Germany          Italy     Yugoslavia        Finland
##     0.13419101     0.02620183     0.16453131     0.20741749     0.40651556
##         Sweden         Norway        Denmark   Sierre Leone          Ghana
##     0.38788418     0.38064494     0.34068209     0.33370269     0.27123052
##          Kenya         Zambia         Malawi   South Africa         Turkey
##     0.25080958     0.22516739     0.36781180     0.15494439     0.19534517
##          Egypt    South Korea          Japan          India       Pakistan
##     0.19039594     0.08886070     0.25827735     0.02282537    -0.09662377
##       Thailand       Malaysia    Philippines      Indonesia      Australia
##     0.24963313     0.28009483     0.13760179     0.25416897     0.21160084
##    New Zealand
##     0.24491843
```

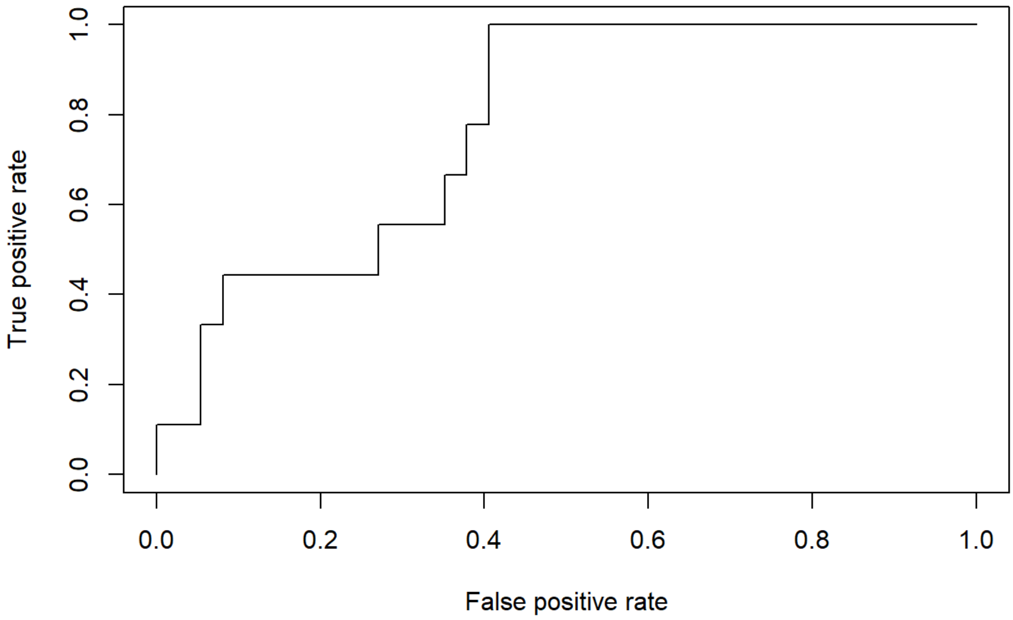
```
length(m3$fitted.values)
```

```
## [1] 46
```

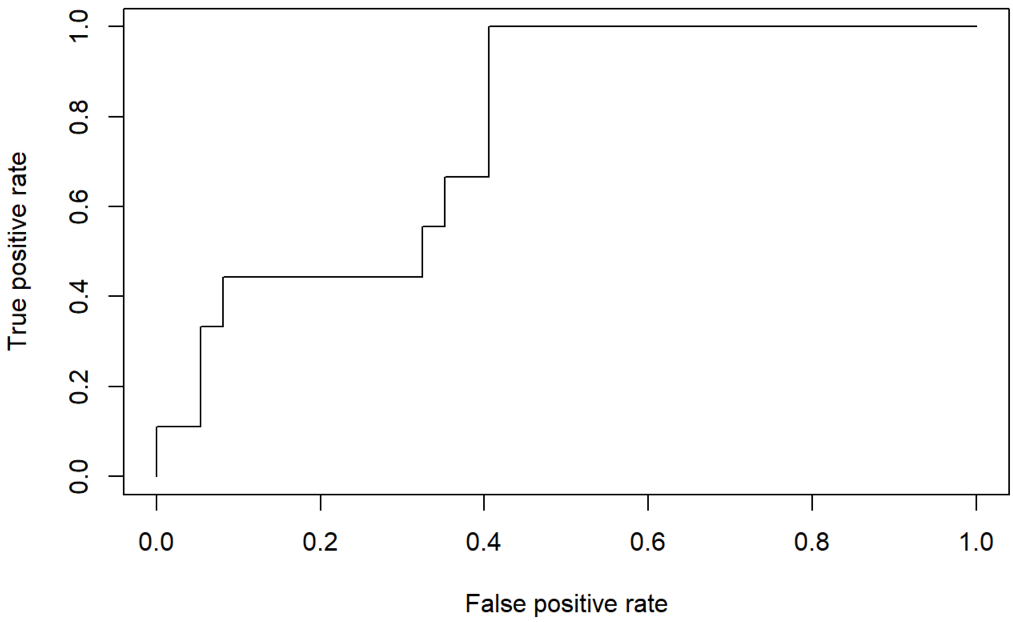
```
## ROC Plots
predm1 <- prediction( m1$fitted.values, ms$deaths)
perfm1 <- performance(predm1,"tpr","fpr")
plot(perfm1)
```




```
predm2 <- prediction( m2$fitted.values, ms$deaths)
perfm2 <- performance(predm2,"tpr","fpr")
plot(perfm2)
```




```
predm3 <- prediction( m3$fitted.values, ms$deaths)
perfm3 <- performance(predm3,"tpr","fpr")
plot(perfm3)
```




```
## Separation Plots
separationplot(pred=as.vector(m1$fitted.values), actual=as.vector(ms$deaths), type="line",line=TRUE, show.ex
pected=TRUE, heading="Separation Plot m1")

separationplot(pred=as.vector(m2$fitted.values), actual=as.vector(ms$deaths), type="line",line=TRUE, show.ex
pected=TRUE, heading="Separation Plot m2")

separationplot(pred=as.vector(m3$fitted.values), actual=as.vector(ms$deaths), type="line",line=TRUE, show.ex
pected=TRUE, heading="Separation Plot m3")
```

Please note: the separation plots will not generate within R markdown for some reason, so I've appended them to the end of this PDF. They're labelled m1-m3.

```r
## Cross-Validation
require(caret)
flds <- createFolds(ms$deaths, k = 10, list = TRUE, returnTrain = FALSE)
perf<-as.list(1:10)
ms$deaths<-as.factor(ms$deaths)
for (i in 1:10)
{test<-ms[flds[[i]],]
train<-ms[-flds[[i]],]
m1 <- glm(as.numeric(deaths) ~ sanctions, data = train)
pred<-predict(m1,type='response',newdata=test)
pred2 <- prediction( pred, as.factor(test$deaths))
perf[[i]] <- performance(pred2,"tpr","fpr")}
```

```
## Error in approxfun(x.values.2, y.values.2, method = "constant", f = 1, : zero non-NA points
```

The binary variable that we created seems to have too many 0 values for a 10-fold cross-validation to work. R is throwing an error whenever one of the sets contains only values of 0. I'll try a different approach:

```r
## Cross-Validation
library(ROCR)
require(caret)
flds <- createFolds(as.factor(ms$deaths), k = 10, list = TRUE, returnTrain = FALSE)
pred<-ms$deaths
ms$deaths<-as.numeric(ms$deaths)
for (i in 1:10)
{test<-ms[flds[[i]],]
train<-ms[-flds[[i]],]
m1 <- glm(deaths ~ sanctions, data = train,family = 'binomial')
pred[flds[[i]]]<-predict(m1,type='response',newdata=data.frame(sanctions=test[,'sanctions']))
}
```

```
## Error in eval(family$initialize): y values must be 0 <= y <= 1
```

```r
pred2 <- prediction( pred, ms$deaths)
```

```
## Error in prediction(pred, ms$deaths): Format of predictions is invalid.
```

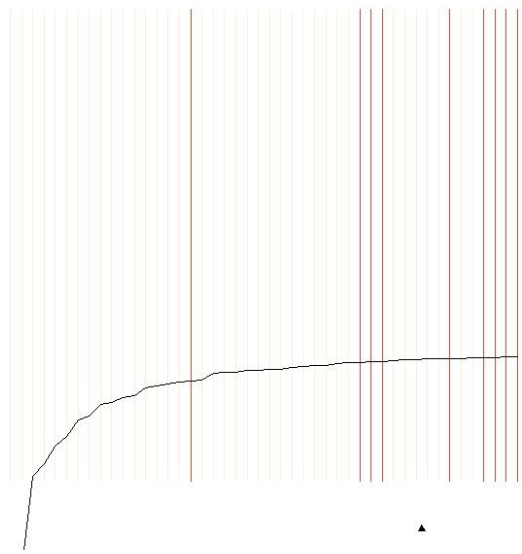
```r
perf<-performance(pred2,"tpr","fpr")
```

```
## Error in approxfun(x.values.2, y.values.2, method = "constant", f = 1, : zero non-NA points
```

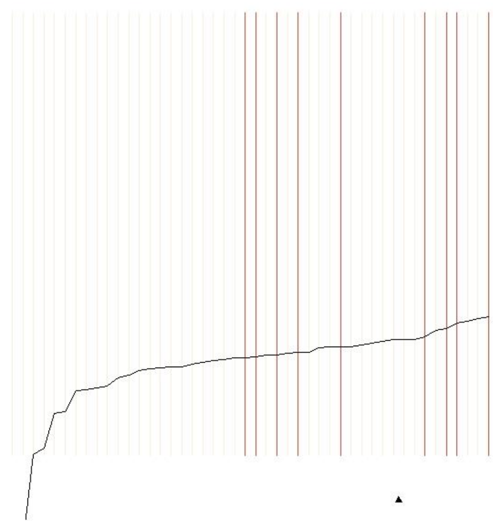
```r
plot(perf)
```

```
## Error in xy.coords(x, y, xlabel, ylabel, log): 'x' is a list, but does not have components 'x' and 'y'
```

Separation Plot m1



Separation Plot m2



Separation Plot m3