Horizontal Inequality, Cross-cutting Cleavages, and Civil War

Replication File

Joshua Gubler
Assistant Professor
Department of Political Science
Brigham Young University[1]

Joel Selway
Assistant Professor
Department of Political Science
Brigham Young University[2]

October 19, 2011

Abstract

This document contains the R-code we used to generate the results in the paper. In an effort to maximize transparency, it is compiled in LaTeX and Sweave. As such, those interested in replication and already running Sweave can simply download the data file as well as the source file for this document, change the file path designated in the line of code to load the data, and then typeset/generate the .pdf with the results. Those not running Sweave can cut and paste the code in this document into R to replicate the results. For those running other stats programs, we have saved the data file in .csv and .dta formats as well.

[1]Contact: jgub@byu.edu.
[2]Contact: joel_selway@byu.edu.

1 To load the data and necessary R packages for the paper

```
> library(multilevel)
> library(xtable)
> library(apsrtable)
> library(ggplot2)
> library(lmtest)
> library(fields)
> library(sandwich)
> library(Zelig)
> load("/Users/jrg27/Dropbox/Selway_Projects/JCR_Oct2011_FinalDocs/jcr_27Oct2010.rdata")
```

**Note:* To load these data for replication, you will need to change the directory path in the last line of the preceding code to reflect the location of the data file on your computer.

2 To generate the data for Table 1

Starting with the first row (PITF), going from left to right

```
> attach(jcr_27Oct2010.df)
> count(CName[ComboCC >= 0 & PITF >= 0])
> stats(PITF[ComboCC >= 0])[2, ]
> stats(PITF[ComboCC >= 0])[3, ]
> count(CName[ComboCC >= 0 & PITF == 1])
> table(CName[ComboCC >= 0 & PITF == 1], PITF[ComboCC >= 0 & PITF ==
+     1])
> stats(EthFrac[ComboCC >= 0 & PITF >= 0])[2, ]
> stats(RelFrac[ComboCC >= 0 & PITF >= 0])[2, ]
```

Second row (PRIO), going from left to right

```
> count(CName[ComboCC >= 0 & PRIO >= 0])
> stats(PRIO[ComboCC >= 0])[2, ]
> stats(PRIO[ComboCC >= 0])[3, ]
> count(CName[ComboCC >= 0 & PRIO == 1])
> table(CName[ComboCC >= 0 & PRIO == 1], PRIO[ComboCC >= 0 & PRIO ==
+     1])
> stats(EthFrac[ComboCC >= 0 & PRIO >= 0])[2, ]
> stats(RelFrac[ComboCC >= 0 & PRIO >= 0])[2, ]
```

Third row (MEPV), going from left to right

```
> count(CName[ComboCC >= 0 & MEPV >= 0])
> stats(MEPV[ComboCC >= 0])[2, ]
> stats(MEPV[ComboCC >= 0])[3, ]
> count(CName[ComboCC >= 0 & MEPV == 1])
> table(CName[ComboCC >= 0 & MEPV == 1], MEPV[ComboCC >= 0 & MEPV ==
+     1])
> stats(EthFrac[ComboCC >= 0 & MEPV >= 0])[2, ]
> stats(RelFrac[ComboCC >= 0 & MEPV >= 0])[2, ]
```

Starting with the fourth row (PITF), going from left to right

```
> count(CName[PITF >= 0])
> stats(PITF)[2, ]
> stats(PITF)[3, ]
> count(CName[PITF == 1])
> table(CName[PITF == 1], PITF[PITF == 1])
> stats(EthFrac[PITF >= 0])[2, ]
> stats(RelFrac[PITF >= 0])[2, ]
```

Fifth row (PRIO), going from left to right

```
> count(CName[PRIO >= 0])
> stats(PRIO)[2, ]
> stats(PRIO)[3, ]
> count(CName[PRIO == 1])
> table(CName[PRIO == 1], PRIO[PRIO == 1])
> stats(EthFrac[PRIO >= 0])[2, ]
> stats(RelFrac[PRIO >= 0])[2, ]
```

Sixth row (MEPV), going from left to right

```
> count(CName[MEPV >= 0])
> stats(MEPV)[2, ]
> stats(MEPV)[3, ]
> count(CName[MEPV == 1])
> table(CName[MEPV == 1], MEPV[MEPV == 1])
> stats(EthFrac[MEPV >= 0])[2, ]
> stats(RelFrac[MEPV >= 0])[2, ]
> detach(jcr_27Oct2010.df)
```

Authorsâ ĂŹ Dataset							
	Total # of Countries	Mean	Stdev	# Countries Experiencing Civil War	# Civil Wars	Mean EthFrac	Mean RelFrac
PITF	80	0.02	0.13	35	59	0.42	0.46
UCDP/PRIO	80	0.01	0.10	19	31	0.42	0.47
MEPV	80	0.01	0.10	22	31	0.42	0.46
Full Dataset							
PITF	159	0.02	0.13	64	110	0.36	0.44
UCDP/PRIO	157	0.01	0.11	49	72	0.37	0.44
MEPV	156	0.01	0.11	55	81	0.36	0.44

Table 1: Summary statistics of countries in our dataset compared to full datasets

3 To generate the data for Table 2

```
> attach(jcr_27Oct2010.df)
> summary.df <- as.data.frame(stats(EIC)[c(2, 3, 4, 8), 1])
> summary.df$EIC <- stats(EIC)[c(2, 3, 4, 8), 1]
> summary.df$ERC <- stats(ERC)[c(2, 3, 4, 8), 1]
> summary.df$EGC <- stats(EGC)[c(2, 3, 4, 8), 1]
> summary.df$ComboCC <- stats(ComboCC)[c(2, 3, 4, 8), 1]
> summary.df$Income <- stats(Income)[c(2, 3, 4, 8), 1]
> summary.df$Oil <- stats(Oil)[c(2, 3, 4, 8), 1]
> summary.df$Population <- stats(Population)[c(2, 3, 4, 8), 1]
> summary.df$Mountains <- stats(Mountains)[c(2, 3, 4, 8), 1]
> summary.df$Instability <- stats(Instability)[c(2, 3, 4, 8), 1]
> detach(jcr_27Oct2010.df)
> summary.tab <- t(summary.df)
> summary.tab <- summary.tab[c(-1), ]
> rm(summary.df)
```

	mean	Std.Dev.	min	max
EIC	0.87	0.07	0.54	1.00
ERC	0.75	0.20	0.05	1.00
EGC	0.69	0.24	0.00	1.00
ComboCC	0.77	0.14	0.37	1.00
Income	7.67	1.05	3.87	11.11
Oil	0.13	0.34	0.00	1.00
Population	9.05	1.46	5.40	14.03
Mountains	2.17	1.41	0.00	4.56
Instability	0.14	0.35	0.00	1.00

Table 2: Summary Statistics

4 To generate the data for the 2x2x2 table in the paper

To calculate the necessary stats

```
> attach(jcr_27Oct2010.df)
> stats(ERC[PITF >= 0])
> stats(EIC[PITF >= 0])
> stats(EGC[PITF >= 0])
> detach(jcr_27Oct2010.df)
```

Code to generate the countries, cell by cell, for the TOP row of the table, left to right

```
> attach(jcr_27Oct2010.df)
> table(CName[ERC > 0.75 & EIC > 0.87 & EGC > 0.69], PITF[ERC >
+     0.75 & EIC > 0.87 & EGC > 0.69])
> table(CName[ERC <= 0.75 & EIC > 0.87 & EGC > 0.69], PITF[ERC <=
+     0.75 & EIC > 0.87 & EGC > 0.69])
> table(CName[ERC > 0.75 & EIC <= 0.87 & EGC > 0.69], PITF[ERC >
+     0.75 & EIC <= 0.87 & EGC > 0.69])
> table(CName[ERC <= 0.75 & EIC <= 0.87 & EGC > 0.69], PITF[ERC <=
+     0.75 & EIC <= 0.87 & EGC > 0.69])
> detach(jcr_27Oct2010.df)
```

Code to generate the countries, cell by cell, for the BOTTOM row of the table, left to right

```
> attach(jcr_27Oct2010.df)
> table(CName[ERC > 0.75 & EIC > 0.87 & EGC <= 0.69], PITF[ERC >
+     0.75 & EIC > 0.87 & EGC <= 0.69])
> table(CName[ERC <= 0.75 & EIC > 0.87 & EGC <= 0.69], PITF[ERC <=
+     0.75 & EIC > 0.87 & EGC <= 0.69])
> table(CName[ERC > 0.75 & EIC <= 0.87 & EGC <= 0.69], PITF[ERC >
+     0.75 & EIC <= 0.87 & EGC <= 0.69])
> table(CName[ERC <= 0.75 & EIC <= 0.87 & EGC <= 0.69], PITF[ERC <=
+     0.75 & EIC <= 0.87 & EGC <= 0.69])
> detach(jcr_27Oct2010.df)
```

5 To generate the regression results in Table 4

Logit estimations, moving from left-to right in the table (using peace spells and splines):

***Note:* We created the splines variables using Tucker's "btscs" program in Stata.

```
> Model1s <- glm(PITF ~ EIC + PITFpeaceyrs + PITFspline1 + PITFspline2 +
+     PITFspline3 + Income + Oil + Population + Mountains + Instability,
+     family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model2s <- glm(PITF ~ EGC + PITFpeaceyrs + PITFspline1 + PITFspline2 +
+     PITFspline3 + Income + Oil + Population + Mountains + Instability,
+     family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model3s <- glm(PITF ~ ERC + PITFpeaceyrs + PITFspline1 + PITFspline2 +
+     PITFspline3 + Income + Oil + Population + Mountains + Instability,
+     family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model4s <- glm(PITF ~ ComboCC + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
```

```
+     Instability, family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model5s <- glm(PITF ~ ComboCC + EthFrac + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
+     Instability, family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model6s <- glm(PITF ~ ComboCC + RelFrac + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
+     Instability, family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model7s <- glm(PITF ~ ComboCC + EthFrac + RelFrac + PITFpeaceyrs +
+     PITFspline1 + PITFspline2 + PITFspline3 + Income + Oil +
+     Population + Mountains + Instability, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model8s <- glm(PRIO ~ ComboCC + PRIOpeaceyrs + PRIOspline1 +
+     PRIOspline2 + PRIOspline3 + Income + Oil + Population + Mountains +
+     Instability, family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model9s <- glm(MEPV ~ ComboCC + MEPVpeaceyrs + MEPVspline1 +
+     MEPVspline2 + MEPVspline3 + Income + Oil + Population + Mountains +
+     Instability, family = binomial(link = "logit"), data = jcr_27Oct2010.df)
```

Clustered Standard Errors R-code

To correct the standard errors to account for panel variation, we wrote the following function (adapted from code originally written by Mahmood Arai) which performs the same function as Stata's "cluster()" command. An updated version of this function as well as code for other helpful R functions can be found at Gubler's academic website.

```
> clse <- function(dat, fm, cluster) {
+     require(sandwich)
+     require(lmtest)
+     not <- attr(fm$model, "na.action")
+     if (!is.null(not)) {
+         cluster <- cluster[-not]
+         dat <- dat[-not, ]
+     }
+     with(dat, {
+         M <- length(unique(cluster))
+         N <- length(cluster)
+         K <- fm$rank
+         dfc <- (M/(M - 1)) * ((N - 1)/(N - K))
+         uj <- apply(estfun(fm), 2, function(x) tapply(x, cluster,
+             sum))
+         vcovCL <- dfc * sandwich(fm, meat = crossprod(uj)/N)
+         coeftest(fm, vcovCL)
+     })
+ }
```

Estimation results with cluster-corrected standard errors (peace spells and splines):

```
> attach(jcr_27Oct2010.df)
> clse(jcr_27Oct2010.df, Model1s, Country)

z test of coefficients:

               Estimate  Std. Error z value  Pr(>|z|)
(Intercept)  -0.80761878  1.61887045 -0.4989 0.6178654
EIC          -3.83263936  1.07435828 -3.5674 0.0003606 ***
PITFpeaceyrs -0.20439649  0.15409725 -1.3264 0.1847032
PITFspline1  -0.00565596  0.00953042 -0.5935 0.5528708
PITFspline2   0.00020361  0.00136813  0.1488 0.8816939
PITFspline3   0.00022340  0.00024627  0.9071 0.3643331
Income       -0.38775235  0.16627164 -2.3320 0.0196985 *
Oil           1.01669286  0.35770093  2.8423 0.0044790 **
Population    0.33643141  0.07581157  4.4377 9.091e-06 ***
Mountains     0.00989868  0.09437954  0.1049 0.9164697
Instability   0.27807509  0.28740515  0.9675 0.3332757
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> clse(jcr_27Oct2010.df, Model2s, Country)

z test of coefficients:

               Estimate  Std. Error z value Pr(>|z|)
(Intercept)  -2.88975483  1.41905365 -2.0364 0.041711 *
EGC          -1.78304835  0.56054472 -3.1809 0.001468 **
PITFpeaceyrs -0.11789102  0.15193118 -0.7760 0.437778
PITFspline1  -0.00067067  0.00937092 -0.0716 0.942944
PITFspline2  -0.00045299  0.00135940 -0.3332 0.738963
PITFspline3   0.00029859  0.00025127  1.1883 0.234709
Income       -0.36126484  0.16072509 -2.2477 0.024594 *
Oil           0.85838598  0.32339623  2.6543 0.007948 **
Population    0.28907516  0.09523622  3.0353 0.002403 **
Mountains     0.02866000  0.09114267  0.3145 0.753178
Instability   0.16614033  0.30612762  0.5427 0.587325
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> clse(jcr_27Oct2010.df, Model3s, Country)

z test of coefficients:

               Estimate  Std. Error z value  Pr(>|z|)
(Intercept)  -2.87252006  1.24301898 -2.3109 0.0208372 *
```

```
ERC          -1.09774360  0.68642663 -1.5992 0.1097729
PITFpeaceyrs -0.25255551  0.16498181 -1.5308 0.1258168
PITFspline1  -0.00811787  0.00958989 -0.8465 0.3972721
PITFspline2   0.00054734  0.00133920  0.4087 0.6827571
PITFspline3   0.00017012  0.00023025  0.7388 0.4600055
Income       -0.36784142  0.14008996 -2.6258 0.0086458 **
Oil           1.02971592  0.34215517  3.0095 0.0026168 **
Population    0.27612564  0.07961568  3.4682 0.0005239 ***
Mountains     0.09415254  0.08889288  1.0592 0.2895230
Instability   0.19419404  0.27322705  0.7107 0.4772439
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> clse(jcr_27Oct2010.df, Model4s, Country)

```
z test of coefficients:

               Estimate  Std. Error z value  Pr(>|z|)
(Intercept)  -1.41301306  1.70190110 -0.8303 0.4063942
ComboCC      -3.63126499  0.99709108 -3.6419 0.0002707 ***
PITFpeaceyrs -0.13782778  0.15413418 -0.8942 0.3712114
PITFspline1  -0.00108981  0.00954260 -0.1142 0.9090755
PITFspline2  -0.00044620  0.00139666 -0.3195 0.7493645
PITFspline3   0.00031423  0.00026476  1.1869 0.2352825
Income       -0.30060822  0.16386156 -1.8345 0.0665760 .
Oil           0.85788309  0.31408885  2.7313 0.0063078 **
Population    0.26161638  0.09567924  2.7343 0.0062512 **
Mountains     0.00707280  0.10013104  0.0706 0.9436879
Instability   0.28062983  0.31873613  0.8804 0.3786180
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> clse(jcr_27Oct2010.df, Model5s, Country)

```
z test of coefficients:

               Estimate  Std. Error z value Pr(>|z|)
(Intercept)  -1.36792240  1.94507163 -0.7033 0.481884
ComboCC      -3.65797808  1.25329142 -2.9187 0.003515 **
EthFrac      -0.03372135  0.75081973 -0.0449 0.964177
PITFpeaceyrs -0.13807012  0.15363749 -0.8987 0.368826
PITFspline1  -0.00111095  0.00959560 -0.1158 0.907829
PITFspline2  -0.00044331  0.00140987 -0.3144 0.753191
PITFspline3   0.00031391  0.00026730  1.1744 0.240234
Income       -0.30358179  0.16444156 -1.8461 0.064872 .
Oil           0.85978873  0.33415806  2.5730 0.010082 *
Population    0.26287011  0.09582235  2.7433 0.006082 **
```

```
Mountains     0.00705840  0.10006412  0.0705 0.943765
Instability   0.28078337  0.31884385  0.8806 0.378518
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> clse(jcr_27Oct2010.df, Model6s, Country)
```

```
z test of coefficients:

               Estimate  Std. Error z value  Pr(>|z|)
(Intercept)  -1.19966074  1.73215629 -0.6926 0.4885717
ComboCC      -3.58205275  0.98252151 -3.6458 0.0002666 ***
RelFrac      -0.38947591  0.61388859 -0.6344 0.5257932
PITFpeaceyrs -0.13803669  0.15391574 -0.8968 0.3698081
PITFspline1  -0.00106543  0.00952063 -0.1119 0.9108966
PITFspline2  -0.00045421  0.00139360 -0.3259 0.7444790
PITFspline3   0.00031707  0.00026489  1.1970 0.2313106
Income       -0.30864679  0.16023657 -1.9262 0.0540801 .
Oil           0.75279336  0.36638309  2.0547 0.0399117 *
Population    0.25949641  0.09475759  2.7385 0.0061715 **
Mountains     0.01235988  0.10048902  0.1230 0.9021092
Instability   0.28583412  0.32072839  0.8912 0.3728203
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> clse(jcr_27Oct2010.df, Model7s, Country)
```

```
z test of coefficients:

               Estimate  Std. Error z value Pr(>|z|)
(Intercept)  -1.35712036  1.96222106 -0.6916 0.489173
ComboCC      -3.47698057  1.26182102 -2.7555 0.005860 **
EthFrac       0.12897294  0.84310379  0.1530 0.878419
RelFrac      -0.41947185  0.68465928 -0.6127 0.540093
PITFpeaceyrs -0.13711088  0.15257345 -0.8987 0.368837
PITFspline1  -0.00098252  0.00951575 -0.1033 0.917763
PITFspline2  -0.00046577  0.00140093 -0.3325 0.739532
PITFspline3   0.00031843  0.00026760  1.1899 0.234073
Income       -0.29760774  0.16917811 -1.7591 0.078554 .
Oil           0.73780746  0.42472867  1.7371 0.082365 .
Population    0.25452405  0.09544192  2.6668 0.007658 **
Mountains     0.01282322  0.10137233  0.1265 0.899339
Instability   0.28565536  0.32087834  0.8902 0.373343
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> clse(jcr_27Oct2010.df, Model8s, Country)
```

```
z test of coefficients:

               Estimate  Std. Error z value Pr(>|z|)
(Intercept)  -3.16933568  2.46813372 -1.2841 0.199106
ComboCC      -2.96045520  1.40226564 -2.1112 0.034756 *
PRIOpeaceyrs -0.18041637  0.10128143 -1.7813 0.074857 .
PRIOspline1  -0.00040558  0.00180700 -0.2244 0.822409
PRIOspline2  -0.00077094  0.00200326 -0.3848 0.700355
PRIOspline3   0.00110413  0.00133284  0.8284 0.407444
Income       -0.28614284  0.21044778 -1.3597 0.173929
Oil           0.44758096  0.34145194  1.3108 0.189920
Population    0.30799723  0.11105299  2.7734 0.005547 **
Mountains     0.29858452  0.14157329  2.1090 0.034941 *
Instability   0.58251477  0.29734953  1.9590 0.050110 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> clse(jcr_27Oct2010.df, Model9s, Country)
```

```
z test of coefficients:

               Estimate  Std. Error z value Pr(>|z|)
(Intercept)  -4.86216441  3.03463284 -1.6022 0.109106
ComboCC      -1.82849709  1.48199593 -1.2338 0.217275
MEPVpeaceyrs  0.13955566  0.32176634  0.4337 0.664494
MEPVspline1   0.02673435  0.01962193  1.3625 0.173049
MEPVspline2  -0.00502300  0.00274354 -1.8308 0.067124 .
MEPVspline3   0.00129873  0.00044641  2.9093 0.003623 **
Income       -0.23822394  0.19366867 -1.2301 0.218675
Oil           0.62909286  0.31248330  2.0132 0.044093 *
Population    0.34890023  0.11453417  3.0463 0.002317 **
Mountains     0.23334153  0.12934540  1.8040 0.071228 .
Instability   0.46120381  0.42534206  1.0843 0.278226
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> detach(jcr_27Oct2010.df)
```

6 To generate the risk ratios in Table 5

To calculate the risk ratios, we use the Zelig program created by Imai et. al. We first re-estimate the models in Table 4 (using GEE, assuming independence within clusters) and then use Zelig's MLE simulation algorithm to run 1000 simulated predictions of Y given the values of X (min/max) we provide. Here is the code for this procedure:

To calculate RR for EIC using Model 1:

```
> Model1z <- zelig(formula = PITF ~ EIC + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
+     Instability, model = "logit.gee", data = jcr_27Oct2010.df,
+     robust = TRUE, id = "Country")
> Model1EIC.high <- setx(Model1z, EIC = 1)
> Model1EIC.low <- setx(Model1z, EIC = 0.538)
> Model1.EIC <- sim(Model1z, x = Model1EIC.high, x1 = Model1EIC.low)
> summary(Model1.EIC)
```

To calculate RR for EGC using Model 2:

```
> Model2z <- zelig(formula = PITF ~ EGC + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
+     Instability, model = "logit.gee", data = jcr_27Oct2010.df,
+     robust = TRUE, id = "Country")
> Model2EGC.high <- setx(Model2z, EGC = 1)
> Model2EGC.low <- setx(Model2z, EGC = 0.538)
> Model2.EGC <- sim(Model2z, x = Model2EGC.high, x1 = Model2EGC.low)
> summary(Model2.EGC)
```

To calculate RR for ERC using Model 3:

```
> Model3z <- zelig(formula = PITF ~ ERC + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
+     Instability, model = "logit.gee", data = jcr_27Oct2010.df,
+     robust = TRUE, id = "Country")
> Model3ERC.high <- setx(Model3z, ERC = 1)
> Model3ERC.low <- setx(Model3z, ERC = 0.538)
> Model3.ERC <- sim(Model3z, x = Model3ERC.high, x1 = Model3ERC.low)
> summary(Model3.ERC)
```

To calculate RR for ComboCC using Model 4:

```
> Model4z <- zelig(formula = PITF ~ ComboCC + PITFpeaceyrs + PITFspline1 +
+     PITFspline2 + PITFspline3 + Income + Oil + Population + Mountains +
+     Instability, model = "logit.gee", data = jcr_27Oct2010.df,
+     robust = TRUE, id = "Country")
> Model4CC.high <- setx(Model4z, ComboCC = 1)
> Model4CC.low <- setx(Model4z, ComboCC = 0.368)
```

```
> Model4.CC <- sim(Model4z, x = Model4CC.high, x1 = Model4CC.low)
> summary(Model4.CC)
```

To calculate RR for Income using Model 4:

```
> Model4Income.high <- setx(Model4z, Income = 11.11)
> Model4Income.low <- setx(Model4z, Income = 3.87)
> Model4.Income <- sim(Model4z, x = Model4Income.high, x1 = Model4Income.low)
> summary(Model4.Income)
```

To calculate RR for Oil using Model 4:

```
> Model4Oil.high <- setx(Model4z, Oil = 1)
> Model4Oil.low <- setx(Model4z, Oil = 0)
> Model4.Oil <- sim(Model4z, x = Model4Oil.high, x1 = Model4Oil.low)
> summary(Model4.Oil)
```

To calculate RR for Population using Model 4:

```
> Model4Population.high <- setx(Model4z, Population = 14.03)
> Model4Population.low <- setx(Model4z, Population = 5.4)
> Model4.Population <- sim(Model4z, x = Model4Population.high,
+     x1 = Model4Population.low)
> summary(Model4.Population)
```

To calculate RR for Mountains using Model 4:

```
> Model4Mountains.high <- setx(Model4z, Mountains = 4.56)
> Model4Mountains.low <- setx(Model4z, Mountains = 0)
> Model4.Mountains <- sim(Model4z, x = Model4Mountains.high, x1 = Model4Mountains.low)
> summary(Model4.Mountains)
```

To calculate RR for Instability using Model 4:

```
> Model4Instability.high <- setx(Model4z, Instability = 1)
> Model4Instability.low <- setx(Model4z, Instability = 0)
> Model4.Instability <- sim(Model4z, x = Model4Instability.high,
+     x1 = Model4Instability.low)
> summary(Model4.Instability)
```

7 Time dummies code and code to generate the plots presented in the online supplementary appendix

Table 4 Estimation results with cluster-corrected standard errors (time dummies):

Code to create the time-dummies for PITF

```
> peacespellPITF <- sort(unique(jcr_27Oct2010.df$PITFpeaceyrs))
> timedummiesPITF <- matrix(NA, nrow = nrow(jcr_27Oct2010.df),
+     ncol = length(peacespellPITF))
> for (j in 1:length(peacespellPITF)) {
+     timedummiesPITF[, j] <- as.integer(jcr_27Oct2010.df$PITFpeaceyrs ==
+         peacespellPITF[j])
+ }
> pyPITF <- c(0:54)
> pyPITF <- paste("psPITF", pyPITF, sep = "")
> dimnames(timedummiesPITF) <- list(1:nrow(timedummiesPITF), pyPITF)
> jcr_27Oct2010.df <- cbind(jcr_27Oct2010.df, timedummiesPITF)
```

Code to create the time-dummies for PRIO

```
> peacespellPRIO <- sort(unique(jcr_27Oct2010.df$PRIOpeaceyrs))
> timedummiesPRIO <- matrix(NA, nrow = nrow(jcr_27Oct2010.df),
+     ncol = length(peacespellPRIO))
> for (j in 1:length(peacespellPRIO)) {
+     timedummiesPRIO[, j] <- as.integer(jcr_27Oct2010.df$PRIOpeaceyrs ==
+         peacespellPRIO[j])
+ }
> pyPRIO <- c(0:49)
> pyPRIO <- paste("psPRIO", pyPRIO, sep = "")
> dimnames(timedummiesPRIO) <- list(1:nrow(timedummiesPRIO), pyPRIO)
> jcr_27Oct2010.df <- cbind(jcr_27Oct2010.df, timedummiesPRIO)
```

Code to create the time-dummies for MEPV

```
> peacespellMEPV <- sort(unique(jcr_27Oct2010.df$MEPVpeaceyrs))
> timedummiesMEPV <- matrix(NA, nrow = nrow(jcr_27Oct2010.df),
+     ncol = length(peacespellMEPV))
> for (j in 1:length(peacespellMEPV)) {
+     timedummiesMEPV[, j] <- as.integer(jcr_27Oct2010.df$MEPVpeaceyrs ==
+         peacespellMEPV[j])
```

```
+ }
> pyMEPV <- c(0:54)
> pyMEPV <- paste("psMEPV", pyMEPV, sep = "")
> dimnames(timedummiesMEPV) <- list(1:nrow(timedummiesMEPV), pyMEPV)
> jcr_27Oct2010.df <- cbind(jcr_27Oct2010.df, timedummiesMEPV)
```

Logit estimations, moving from left-to right in the table (using time dummies):

```
> Model1 <- glm(PITF ~ EIC + Income + Oil + Population + Mountains +
+     Instability + psPITF1 + psPITF2 + psPITF3 + psPITF4 + psPITF5 +
+     psPITF6 + psPITF7 + psPITF8 + psPITF9 + psPITF10 + psPITF11 +
+     psPITF12 + psPITF13 + psPITF14 + psPITF15 + psPITF16 + psPITF17 +
+     psPITF18 + psPITF19 + psPITF20 + psPITF21 + psPITF22 + psPITF23 +
+     psPITF24 + psPITF25 + psPITF26 + psPITF27 + psPITF28 + psPITF29 +
+     psPITF30 + psPITF31 + psPITF32 + psPITF33 + psPITF34 + psPITF35 +
+     psPITF36 + psPITF37 + psPITF38 + psPITF39 + psPITF40 + psPITF41 +
+     psPITF42 + psPITF43 + psPITF44 + psPITF45 + psPITF46 + psPITF47 +
+     psPITF48 + psPITF49 + psPITF50, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model2 <- glm(PITF ~ EGC + Income + Oil + Population + Mountains +
+     Instability + psPITF1 + psPITF2 + psPITF3 + psPITF4 + psPITF5 +
+     psPITF6 + psPITF7 + psPITF8 + psPITF9 + psPITF10 + psPITF11 +
+     psPITF12 + psPITF13 + psPITF14 + psPITF15 + psPITF16 + psPITF17 +
+     psPITF18 + psPITF19 + psPITF20 + psPITF21 + psPITF22 + psPITF23 +
+     psPITF24 + psPITF25 + psPITF26 + psPITF27 + psPITF28 + psPITF29 +
+     psPITF30 + psPITF31 + psPITF32 + psPITF33 + psPITF34 + psPITF35 +
+     psPITF36 + psPITF37 + psPITF38 + psPITF39 + psPITF40 + psPITF41 +
+     psPITF42 + psPITF43 + psPITF44 + psPITF45 + psPITF46 + psPITF47 +
+     psPITF48 + psPITF49 + psPITF50, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model3 <- glm(PITF ~ ERC + Income + Oil + Population + Mountains +
+     Instability + psPITF1 + psPITF2 + psPITF3 + psPITF4 + psPITF5 +
+     psPITF6 + psPITF7 + psPITF8 + psPITF9 + psPITF10 + psPITF11 +
+     psPITF12 + psPITF13 + psPITF14 + psPITF15 + psPITF16 + psPITF17 +
+     psPITF18 + psPITF19 + psPITF20 + psPITF21 + psPITF22 + psPITF23 +
+     psPITF24 + psPITF25 + psPITF26 + psPITF27 + psPITF28 + psPITF29 +
+     psPITF30 + psPITF31 + psPITF32 + psPITF33 + psPITF34 + psPITF35 +
+     psPITF36 + psPITF37 + psPITF38 + psPITF39 + psPITF40 + psPITF41 +
+     psPITF42 + psPITF43 + psPITF44 + psPITF45 + psPITF46 + psPITF47 +
+     psPITF48 + psPITF49 + psPITF50, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model4 <- glm(PITF ~ ComboCC + Income + Oil + Population + Mountains +
+     Instability + psPITF1 + psPITF2 + psPITF3 + psPITF4 + psPITF5 +
+     psPITF6 + psPITF7 + psPITF8 + psPITF9 + psPITF10 + psPITF11 +
```

```
+     psPITF12 + psPITF13 + psPITF14 + psPITF15 + psPITF16 + psPITF17 +
+     psPITF18 + psPITF19 + psPITF20 + psPITF21 + psPITF22 + psPITF23 +
+     psPITF24 + psPITF25 + psPITF26 + psPITF27 + psPITF28 + psPITF29 +
+     psPITF30 + psPITF31 + psPITF32 + psPITF33 + psPITF34 + psPITF35 +
+     psPITF36 + psPITF37 + psPITF38 + psPITF39 + psPITF40 + psPITF41 +
+     psPITF42 + psPITF43 + psPITF44 + psPITF45 + psPITF46 + psPITF47 +
+     psPITF48 + psPITF49 + psPITF50, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model5 <- glm(PITF ~ ComboCC + EthFrac + Income + Oil + Population +
+     Mountains + Instability + psPITF1 + psPITF2 + psPITF3 + psPITF4 +
+     psPITF5 + psPITF6 + psPITF7 + psPITF8 + psPITF9 + psPITF10 +
+     psPITF11 + psPITF12 + psPITF13 + psPITF14 + psPITF15 + psPITF16 +
+     psPITF17 + psPITF18 + psPITF19 + psPITF20 + psPITF21 + psPITF22 +
+     psPITF23 + psPITF24 + psPITF25 + psPITF26 + psPITF27 + psPITF28 +
+     psPITF29 + psPITF30 + psPITF31 + psPITF32 + psPITF33 + psPITF34 +
+     psPITF35 + psPITF36 + psPITF37 + psPITF38 + psPITF39 + psPITF40 +
+     psPITF41 + psPITF42 + psPITF43 + psPITF44 + psPITF45 + psPITF46 +
+     psPITF47 + psPITF48 + psPITF49 + psPITF50, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model6 <- glm(PITF ~ ComboCC + RelFrac + Income + Oil + Population +
+     Mountains + Instability + psPITF1 + psPITF2 + psPITF3 + psPITF4 +
+     psPITF5 + psPITF6 + psPITF7 + psPITF8 + psPITF9 + psPITF10 +
+     psPITF11 + psPITF12 + psPITF13 + psPITF14 + psPITF15 + psPITF16 +
+     psPITF17 + psPITF18 + psPITF19 + psPITF20 + psPITF21 + psPITF22 +
+     psPITF23 + psPITF24 + psPITF25 + psPITF26 + psPITF27 + psPITF28 +
+     psPITF29 + psPITF30 + psPITF31 + psPITF32 + psPITF33 + psPITF34 +
+     psPITF35 + psPITF36 + psPITF37 + psPITF38 + psPITF39 + psPITF40 +
+     psPITF41 + psPITF42 + psPITF43 + psPITF44 + psPITF45 + psPITF46 +
+     psPITF47 + psPITF48 + psPITF49 + psPITF50, family = binomial(link = "logit"),
+     data = jcr_27Oct2010.df)
> Model7 <- glm(PITF ~ ComboCC + EthFrac + RelFrac + Income + Oil +
+     Population + Mountains + Instability + psPITF1 + psPITF2 +
+     psPITF3 + psPITF4 + psPITF5 + psPITF6 + psPITF7 + psPITF8 +
+     psPITF9 + psPITF10 + psPITF11 + psPITF12 + psPITF13 + psPITF14 +
+     psPITF15 + psPITF16 + psPITF17 + psPITF18 + psPITF19 + psPITF20 +
+     psPITF21 + psPITF22 + psPITF23 + psPITF24 + psPITF25 + psPITF26 +
+     psPITF27 + psPITF28 + psPITF29 + psPITF30 + psPITF31 + psPITF32 +
+     psPITF33 + psPITF34 + psPITF35 + psPITF36 + psPITF37 + psPITF38 +
+     psPITF39 + psPITF40 + psPITF41 + psPITF42 + psPITF43 + psPITF44 +
+     psPITF45 + psPITF46 + psPITF47 + psPITF48 + psPITF49 + psPITF50,
+     family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model8 <- glm(PRIO ~ ComboCC + Income + Oil + Population + Mountains +
+     Instability + psPRIO1 + psPRIO2 + psPRIO3 + psPRIO4 + psPRIO5 +
+     psPRIO6 + psPRIO7 + psPRIO8 + psPRIO9 + psPRIO10 + psPRIO11 +
+     psPRIO12 + psPRIO13 + psPRIO14 + psPRIO15 + psPRIO16 + psPRIO17 +
```

```
+     psPRIO18 + psPRIO19 + psPRIO20 + psPRIO21 + psPRIO22 + psPRIO23 +
+     psPRIO24 + psPRIO25 + psPRIO26 + psPRIO27 + psPRIO28 + psPRIO29 +
+     psPRIO30 + psPRIO31 + psPRIO32 + psPRIO33 + psPRIO34 + psPRIO35 +
+     psPRIO36 + psPRIO37 + psPRIO38 + psPRIO39 + psPRIO40 + psPRIO41 +
+     psPRIO42 + psPRIO43 + psPRIO44 + psPRIO45 + psPRIO46 + psPRIO47 +
+     psPRIO48 + psPRIO49, family = binomial(link = "logit"), data = jcr_27Oct2010.df)
> Model9 <- glm(MEPV ~ ComboCC + Income + Oil + Population + Mountains +
+     Instability + psMEPV1 + psMEPV2 + psMEPV3 + psMEPV4 + psMEPV5 +
+     psMEPV6 + psMEPV7 + psMEPV8 + psMEPV9 + psMEPV10 + psMEPV11 +
+     psMEPV12 + psMEPV13 + psMEPV14 + psMEPV15 + psMEPV16 + psMEPV17 +
+     psMEPV18 + psMEPV19 + psMEPV20 + psMEPV21 + psMEPV22 + psMEPV23 +
+     psMEPV24 + psMEPV25 + psMEPV26 + psMEPV27 + psMEPV28 + psMEPV29 +
+     psMEPV30 + psMEPV31 + psMEPV32 + psMEPV33 + psMEPV34 + psMEPV35 +
+     psMEPV36 + psMEPV37 + psMEPV38 + psMEPV39 + psMEPV40 + psMEPV41 +
+     psMEPV42 + psMEPV43 + psMEPV44 + psMEPV45 + psMEPV46 + psMEPV47 +
+     psMEPV48 + psMEPV49 + psMEPV50 + psMEPV51 + psMEPV52 + psMEPV53 +
+     psMEPV54, family = binomial(link = "logit"), data = jcr_27Oct2010.df)

> attach(jcr_27Oct2010.df)
> clse(jcr_27Oct2010.df, Model1, Country)
> clse(jcr_27Oct2010.df, Model2, Country)
> clse(jcr_27Oct2010.df, Model3, Country)
> clse(jcr_27Oct2010.df, Model4, Country)
> clse(jcr_27Oct2010.df, Model5, Country)
> clse(jcr_27Oct2010.df, Model6, Country)
> clse(jcr_27Oct2010.df, Model7, Country)
> clse(jcr_27Oct2010.df, Model8, Country)
> clse(jcr_27Oct2010.df, Model9, Country)
> detach(jcr_27Oct2010.df)
```

To generate the scatter plots at the start of the supplementary appendix

Code for the Ethnic Fractionalization and ERC plot:

```
> attach(jcr_27Oct2010.df)
> plot(EthFrac, ERC, type = "n")
> text(EthFrac, ERC, labels = CName)
> detach(jcr_27Oct2010.df)
```

Code for the Ethnic Fractionalization and EGC plot:

```
> attach(jcr_27Oct2010.df)
> plot(EthFrac, EGC, type = "n")
```

```
> text(EthFrac, EGC, labels = CName)
> detach(jcr_27Oct2010.df)
```

Code for the Ethnic Fractionalization and EIC plot:

```
> attach(jcr_27Oct2010.df)
> plot(EthFrac, EIC, type = "n")
> text(EthFrac, EIC, labels = CName)
> detach(jcr_27Oct2010.df)
```

****A concluding note:* We present the raw results for the robustness checks and PCA analysis (both conducted in Stata) in the online supplementary appendix itself. As such, we do not duplicate the code and results here. Please contact the authors with any additional questions.