

```

1
2 ///////////////////////////////////////////////////////////////////
3 //                MCC FY21 Data Cleaning File                //
4 //                S&E Team                                    //
5 ///////////////////////////////////////////////////////////////////
6
7 /*
8 The following file provides a step by step demonstration of the
9 process used by MCC's S&E Team to clean all of the raw data files
10 from indicator institutions and create the scorecards.  These steps
11 are outlined in MCC's FY21 Guide to the Indicators
12 (https://www.mcc.gov/resources/doc/guide-to-the-indicators-fy-2021)
13 This file is included to provide complete clarity as to exactly how
14 these indicators are constructed.
15
16 To use this file, you need to download the appropriate files from the
17 indicator institutions as described below, and set the global macros
18 to the correct locations.  Note that many of MCC's indicator institutions
19 historically revise their data, so newly downloaded data may no longer match
20 the data recieved by MCC.  MCC does not historically revised scorecard data,
21 so the data presented in the Open Data is a record of the data produced by
22 that institution at the time of scorecard publication.
23
24 This file will output individual files for each indicator.  You can run
25 sections of the program individually, but the section on GNI must be run
26 for the rest to work.
27
28 The official final data files are contained on MCC's open data site.
29 The purpose of this file is to provide systematic instructions explaining
30 the methodology used to create these files.  Due to methodological changes
31 and historical data revisions by indicator institutions or MCC, this program
32 is not designed to recreate previous year's scorecards.  Rather it is designed
33 to clearly show MCC's methodology in FY21.  Updates will be made annually as needed.
34 */
35
36 *****
37 ** Preliminary Settings **
38 *****
39 ** The following settings set up the necessary environment for the do file to run
40 ** This includes programs which are run repeatedly in the file.
41
42 set more off
43 clear
44 ///////////////////////////////////////////////////////////////////
45 // Global Macros //
46 ///////////////////////////////////////////////////////////////////
47
48 //To use this file you need to set the "globals" to the correct file paths on your computer
49 //raw_data is where you are storing the raw data, (instructions for download are included
50 in each section below)
51 //created_data is where you would like the program to store temporary files
52 //final_data is where you would like the program to output the final files
53 global raw_data "WRITE_RAW_FILE_PATH_HERE" //Your filepath will probably be something like
54 C:\Users\YOURUSERNAME\Documents\RAWDATAFOLDERNAME\
55 global created_data "WRITE_CREATED_FILE_PATH_HERE" //Your filepath will probably be
56 something like C:\Users\YOURUSERNAME\Documents\CREATEDDATAFOLDERNAME\
57 global final_data "WRITE_FINAL_FILE_PATH_HERE" //Your filepath will probably be something
58 like C:\Users\YOURUSERNAME\Documents\FINALDATAFOLDERNAME\
59
60 ///////////////////////////////////////////////////////////////////
61 // Programs //
62 ///////////////////////////////////////////////////////////////////
63 //The following sub-programs are used throughout the main program
64 //to rename countries and calculate percentiles.
65
66 **WBNaming
67 *****
68 //The following program is used to rename countries to a common set of names

```

```

67 //So that they can easily be merged together.
68
69 capture program drop WBNaming
70 program WBNaming
71 args country_variable
72
73 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Côte d'Ivoire"
74 replace `country_variable' = "Sao Tome and Principe" if `country_variable' == "São Tomé and
Principe"
75 replace `country_variable' = "Israeli Occupied Territories*" if `country_variable' ==
"Israeli Occupied Territories*" | `country_variable' == "Israeli-Occupied Territories*"
76 replace `country_variable' = "Kosovo" if `country_variable' == "Kosovo*"
77 replace `country_variable' = "Chechnya" if `country_variable' == "Chechnya*"
78 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote dâ€™Ivoire"
79 replace `country_variable' = "Cyprus" if `country_variable' == "Cyprus "
80 replace `country_variable' = "Hong Kong*" if `country_variable' == "Hong Kong*"
81 replace `country_variable' = "Nagorno-Karabakh*" if `country_variable' ==
"Nagorno-Karabakh*"
82 replace `country_variable' = "Northern Cyprus" if `country_variable' == "Northern Cyprus*"
83 replace `country_variable' = "Pakistani Kashmir" if `country_variable' == "Pakistani
Kashmir*"
84 replace `country_variable' = "Palestinian Authority Administered Territories*" if
`country_variable' == "Palestinian Authority Administered Territories*" |
`country_variable' == "Palestinian Authority-Administered Territories*"
85 replace `country_variable' = "Puerto Rico" if `country_variable' == "Puerto Rico*"
86 replace `country_variable' = "South Ossetia" if `country_variable' == "South Ossetia*"
87 replace `country_variable' = "Transnistria" if `country_variable' == "Transnistria*"
88 replace `country_variable' = "Bahamas, The" if `country_variable' == "Bahamas"
89 replace `country_variable' = "Brunei Darussalam" if `country_variable' == "Brunei"
90 replace `country_variable' = "Cabo Verde" if `country_variable' == "Cape Verde"
91 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo (Brazzaville)"
92 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo (Kinshasa)"
93 replace `country_variable' = "Egypt, Arab Rep." if `country_variable' == "Egypt"
94 replace `country_variable' = "Iran, Islamic Rep." if `country_variable' == "Iran"
95 replace `country_variable' = "Kyrgyz Republic" if `country_variable' == "Kyrgyzstan"
96 replace `country_variable' = "Lao PDR" if `country_variable' == "Laos"
97 replace `country_variable' = "North Macedonia" if `country_variable' == "Macedonia"
98 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' == "Micronesia"
99 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "North
Korea"
100 replace `country_variable' = "Russian Federation" if `country_variable' == "Russia"
101 replace `country_variable' = "Slovak Republic" if `country_variable' == "Slovakia"
102 replace `country_variable' = "Korea, Rep." if `country_variable' == "South Korea"
103 replace `country_variable' = "Syrian Arab Republic" if `country_variable' == "Syria"
104 replace `country_variable' = "Gambia, The" if `country_variable' == "The Gambia"
105 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela"
106 replace `country_variable' = "Yemen, Rep." if `country_variable' == "Yemen"
107 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Côte d'Ivoire"
108 replace `country_variable' = "Bahamas, The" if `country_variable' == "Bahamas"
109 replace `country_variable' = "Bolivia" if `country_variable' == "Bolivia (Plurinational
State of)"
110 replace `country_variable' = "Hong Kong SAR, China" if `country_variable' == "China, Hong
Kong Special Administrative Region"
111 replace `country_variable' = "Macao SAR, China" if `country_variable' == "China, Macao
Special Administrative Region"
112 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo"
113 replace `country_variable' = "Czech Republic" if `country_variable' == "Czechia"
114 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' ==
"Democratic People's Republic of Korea"
115 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Democratic
Republic of the Congo"
116 replace `country_variable' = "Egypt, Arab Rep." if `country_variable' == "Egypt"
117 replace `country_variable' = "Gambia, The" if `country_variable' == "Gambia"
118 replace `country_variable' = "Iran, Islamic Rep." if `country_variable' == "Iran (Islamic
Republic of)"
119 replace `country_variable' = "Kyrgyz Republic" if `country_variable' == "Kyrgyzstan"
120 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao People's Democratic
Republic"
121 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' == "Micronesia
(Federated States of)"

```

```

122 replace `country_variable' = "Korea, Rep." if `country_variable' == "Republic of Korea"
123 replace `country_variable' = "Moldova" if `country_variable' == "Republic of Moldova"
124 replace `country_variable' = "St. Kitts and Nevis" if `country_variable' == "Saint Kitts
and Nevis"
125 replace `country_variable' = "St. Lucia" if `country_variable' == "Saint Lucia"
126 replace `country_variable' = "St. Vincent and the Grenadines" if `country_variable' ==
"Saint Vincent and the Grenadines"
127 replace `country_variable' = "Slovak Republic" if `country_variable' == "Slovakia"
128 replace `country_variable' = "United Kingdom" if `country_variable' == "United Kingdom of
Great Britain and Northern Ireland"
129 replace `country_variable' = "Tanzania" if `country_variable' == "United Republic of
Tanzania"
130 replace `country_variable' = "United States" if `country_variable' == "United States of
America"
131 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela
(Bolivarian Republic of)"
132 replace `country_variable' = "Vietnam" if `country_variable' == "Viet Nam"
133 replace `country_variable' = "Yemen, Rep." if `country_variable' == "Yemen"
134 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Palestine"
135 replace `country_variable' = "Eswatini" if `country_variable' == "Swaziland"
136 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "CÔte d'Ivoire"
137 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "CÃte d'Ivoire"
138 replace `country_variable' = "Syrian Arab Republic" if `country_variable' == "Syrian Arab
Republic (the)"
139 replace `country_variable' = "Bahamas, The" if `country_variable' == "Bahamas (the)"
140 replace `country_variable' = "United Kingdom" if `country_variable' == "United Kingdom of
Great Britain and Northern Ireland (the)"
141 replace `country_variable' = "Central African Republic" if `country_variable' == "Central
African Republic (the)"
142 replace `country_variable' = "Comoros" if `country_variable' == "Comoros (the)"
143 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo (the)"
144 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' ==
"Democratic People's Republic of Korea (the)"
145 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Democratic
Republic of the Congo (the)"
146 replace `country_variable' = "Dominican Republic" if `country_variable' == "Dominican
Republic (the)"
147 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao People's Democratic
Republic (the)"
148 replace `country_variable' = "Marshall Islands" if `country_variable' == "Marshall Islands
(the)"
149 replace `country_variable' = "Netherlands" if `country_variable' == "Netherlands (the)"
150 replace `country_variable' = "Niger" if `country_variable' == "Niger (the)"
151 replace `country_variable' = "Philippines" if `country_variable' == "Philippines (the)"
152 replace `country_variable' = "Korea, Rep." if `country_variable' == "Republic of Korea (the)"
153 replace `country_variable' = "Moldova" if `country_variable' == "Republic of Moldova (the)"
154 replace `country_variable' = "Russian Federation" if `country_variable' == "Russian
Federation (the)"
155 replace `country_variable' = "Sudan" if `country_variable' == "Sudan (the)"
156 replace `country_variable' = "United Arab Emirates" if `country_variable' == "United Arab
Emirates (the)"
157 replace `country_variable' = "United States" if `country_variable' == "United States of
America (the)"
158 replace `country_variable' = "Hong Kong SAR, China" if `country_variable' == "Hong Kong"
159 replace `country_variable' = "Sao Tome and Principe" if `country_variable' == "SÃo Tomé and
Príncipe"
160 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
Dem. Rep."
161 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, The
Democratic Republic of the"
162 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo, Rep. of"
163 replace `country_variable' = "Tanzania" if `country_variable' == "Tanzania, United Republic
of"
164 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
Democratic People's Republic of"
165 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' == "Micronesia,
Federated States of"
166 replace `country_variable' = "Bolivia" if `country_variable' == "Bolivia, Plurinational
State of"
167 replace `country_variable' = "Moldova" if `country_variable' == "Moldova, Republic of"

```

```

168 replace `country_variable' = "Bosnia and Herzegovina" if `country_variable' == "Bosnia
Herzegovina"
169 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao People's Dem. Rep."
170 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Democratic
Republic"
171 replace `country_variable' = "Curaçao" if `country_variable' == "Curacao"
172 replace `country_variable' = "Timor-Leste" if `country_variable' == "East Timor"
173 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' == "Federated
States of Micronesia"
174 replace `country_variable' = "Kazakhstan" if `country_variable' == "Kazakstan"
175 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
Dem. People's Rep. of"
176 replace `country_variable' = "Libya" if `country_variable' == "Libyan Arab Jamahiriya"
177 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Palestinian
Territory, Occupied"
178 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "State of
Palestine"
179 replace `country_variable' = "Reunion" if `country_variable' == "Réunion"
180 replace `country_variable' = "Saint-Barthelemy" if `country_variable' == "Saint-Barthélemy"
181 replace `country_variable' = "St. Martin (French part)" if `country_variable' ==
"Saint-Martin (French)"
182 replace `country_variable' = "St. Martin (French part)" if `country_variable' ==
"Saint-Martin (French part)"
183 replace `country_variable' = "Samoa" if `country_variable' == "Western Samoa"
184 replace `country_variable' = "North Macedonia" if `country_variable' == "The former
Yugoslav Rep. of Macedonia"
185 replace `country_variable' = "Tanzania" if `country_variable' == "United Rep. of Tanzania"
186 replace `country_variable' = "Faroe Islands" if `country_variable' == "Faeroe Islands"
187 replace `country_variable' = "North Macedonia" if `country_variable' == "The former
Yugoslav Republic of Macedonia"
188 replace `country_variable' = "Korea, Rep." if `country_variable' == "Korea, Republic of"
189 replace `country_variable' = "Macao SAR, China" if `country_variable' == "China Macao
Special Administrative Region"
190 replace `country_variable' = "Hong Kong SAR, China" if `country_variable' == "China Hong
Kong Special Administrative Region"
191 replace `country_variable' = "Virgin Islands (U.S.)" if `country_variable' == "United
States Virgin Islands"
192 replace `country_variable' = "Reunion" if `country_variable' == "RÃ©union"
193 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo, Republic of the"
194 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Democratic
Republic of"
195 replace `country_variable' = "Gambia, The" if `country_variable' == "Gambia (The)"
196 replace `country_variable' = "Central African Republic" if `country_variable' == "Central
African Rep"
197 replace `country_variable' = "Cabo Verde" if `country_variable' == "Cabo Verde Republic of"
198 replace `country_variable' = "Bolivia" if `country_variable' == "Bolivia Plurinational
States of "
199 replace `country_variable' = "Myanmar" if `country_variable' == "Burma"
200 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Palestinian
Authority Administered Territories*"
201 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Palestinian
Territories"
202 replace `country_variable' = "Caribbean Netherlands" if `country_variable' == "Bonaire
Saint Eustatius and Saba"
203 replace `country_variable' = "South Sudan" if `country_variable' == "Republic of South Sudan"
204 replace `country_variable' = "Lebanon" if `country_variable' == "LebaNon"
205 replace `country_variable' = "Puerto Rico" if `country_variable' == "Puerto Rico (U.S.)"
206 replace `country_variable' = "North Macedonia" if `country_variable' == "Macedonia, FYR"
207 replace `country_variable' = "Taiwan" if `country_variable' == "Taiwan, China"
208 replace `country_variable' = "San Marino" if `country_variable' == "San MariNo"
209 replace `country_variable' = "Timor-Leste" if `country_variable' == "Timor Leste"
210 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Occupied
Palestinian Territory"
211 replace `country_variable' = "Czech Republic" if `country_variable' == "Czechia"
212 replace `country_variable' = "Guinea-Bissau" if `country_variable' == "Guinea Bissau"
213 replace `country_variable' = "Guinea-Bissau" if `country_variable' == "GuineaBissau"
214 replace `country_variable' = "North Macedonia" if `country_variable' == "The former
Yugoslav Republic of Macedonia"
215 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "State of
Palestine"

```

```

216 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "West Bank and
    Gaza Strip"
217 replace `country_variable' = "Timor-Leste" if `country_variable' == "TimorLeste"
218 replace `country_variable' = "United States" if `country_variable' == "United States of
    America"
219 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela
    (Bolivarian Republic of)**"
220 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela
    (Bolivarian Republic of)"
221 replace `country_variable' = "Northern Cyprus" if `country_variable' == "Northern Cyprus*"
222 replace `country_variable' = "Northern Cyprus" if `country_variable' == "Cyprus (Turkish)"
223 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Ivory Coast"
224 replace `country_variable' = "Iran, Islamic Rep." if `country_variable' == "Islamic
    Republic of Iran"
225 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "The Democratic
    Republic Of The Congo"
226 replace `country_variable' = "Northern Cyprus" if `country_variable' == "Cyprus North"
227 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
    North"
228 replace `country_variable' = "Korea, Rep." if `country_variable' == "Korea, South"
229 replace `country_variable' = "Macao SAR, China" if `country_variable' == "Macau"
230 replace `country_variable' = "Congo, Rep." if `country_variable' == "Republic of the Congo
    (Brazzaville)"
231 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Dem. Rep. of the
    Congo"
232 replace `country_variable' = "Sao Tome and Principe" if `country_variable' == "São Tomé e
    Príncipe"
233 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Dem. Rep. of Congo"
234 replace `country_variable' = "Antigua and Barbuda" if `country_variable' == "Antigua"
235 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote d'Ivoire"
236 replace `country_variable' = "Tibet" if `country_variable' == "Tibet*"
237 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
    Dem. People's Rep."
238 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Democratic
    Republic of the Congo"
239 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo, Republic of"
240 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
    North "
241 replace `country_variable' = "Taiwan" if `country_variable' == "Taiwan "
242 replace `country_variable' = "Eswatini" if `country_variable' == "eSwatini"
243 replace `country_variable' = "Bosnia and Herzegovina" if `country_variable' == "Bosnia"
244 replace `country_variable' = "Bosnia and Herzegovina" if `country_variable' == "Bosnia &
    Herz."
245 replace `country_variable' = "Bosnia and Herzegovina" if `country_variable' == "Bosnia And
    Herzegovina"
246 replace `country_variable' = "Bosnia and Herzegovina" if `country_variable' ==
    "Bosnia-Herzegovina"
247 replace `country_variable' = "Myanmar" if `country_variable' == "Burma (Myanmar)"
248 replace `country_variable' = "Myanmar" if `country_variable' == "Burma/Myanmar"
249 replace `country_variable' = "Central African Republic" if `country_variable' == "Central
    Afr. Rep."
250 replace `country_variable' = "China" if `country_variable' == "China Mainland"
251 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Dem. Rep.
    (Zaire)"
252 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Democratic
    Republic Of (Kinshasa)"
253 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo, Rep"
254 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo, RepublicOf
    (Brazzaville)"
255 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote D'Ivoire"
256 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote D'Ivoire"
257 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote d'Ivoire"
258 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote d'Ivoire"
259 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Côte D'Ivoire"
260 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote d'Ivoire"
261 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Cote d'Ivoire"
262 replace `country_variable' = "Cyprus" if `country_variable' == "Cyprus (Greek)"
263 replace `country_variable' = "Czech Republic" if `country_variable' == "Czech Rep."
264 replace `country_variable' = "Czech Republic" if `country_variable' == "Czech Republic"
265 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Dem. Rep. Congo"
    
```

```

266 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Democratic
Republic Of Congo"
267 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Democratic
Republic Of The Congo"
268 replace `country_variable' = "Vietnam" if `country_variable' == "Democratic Republic Of
Vietnam"
269 replace `country_variable' = "Dominican Republic" if `country_variable' == "Dom. Rep."
270 replace `country_variable' = "Dominican Republic" if `country_variable' == "Dominican Rep."
271 replace `country_variable' = "Dominican Republic" if `country_variable' == "Dominican
Republic"
272 replace `country_variable' = "Guinea-Bissau" if `country_variable' == "Guinea Bissao"
273 replace `country_variable' = "Hong Kong SAR, China" if `country_variable' == "Hong Kong
Sar, China"
274 replace `country_variable' = "West Bank and Gaza" if `country_variable' ==
"Israeli-Occupied Territories And Palestinian Authority"
275 replace `country_variable' = "Korea, Rep." if `country_variable' == "Korea"
276 replace `country_variable' = "Kyrgyz Republic" if `country_variable' == "Kyrgyz Rep."
277 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao PDR"
278 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao Pdr"
279 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao, Pdr"
280 replace `country_variable' = "Macao SAR, China" if `country_variable' == "Macao"
281 replace `country_variable' = "North Macedonia" if `country_variable' == "Macedonia, Fyr"
282 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' == "Micronesia,
Fs"
283 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Palestine/Gaza"
284 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "Palestine/West
Bank"
285 replace `country_variable' = "Congo, Rep." if `country_variable' == "Republic Of The Congo"
286 replace `country_variable' = "Romania" if `country_variable' == "Roumania"
287 replace `country_variable' = "St. Vincent and the Grenadines" if `country_variable' ==
"Saint Vincent And The Grenadine"
288 replace `country_variable' = "St. Vincent and the Grenadines" if `country_variable' ==
"Saint Vincent And The Grenadines"
289 replace `country_variable' = "Sao Tome and Principe" if `country_variable' == "Sao Tome And
Pr."
290 replace `country_variable' = "Serbia & Montenegro" if `country_variable' == "Serbia And
Montenegro"
291 replace `country_variable' = "St. Kitts and Nevis" if `country_variable' == "St. Kitts And
Nev."
292 replace `country_variable' = "St. Vincent and the Grenadines" if `country_variable' == "St.
Vincent And The Grenadines"
293 replace `country_variable' = "South Sudan" if `country_variable' == "Sudan South"
294 replace `country_variable' = "Sao Tome and Principe" if `country_variable' == "SãO Tomé And
Principe"
295 replace `country_variable' = "Trinidad And Tobago" if `country_variable' == "Trinidad And
Tob."
296 replace `country_variable' = "United Arab Emirates" if `country_variable' == "UAE"
297 replace `country_variable' = "United Arab Emirates" if `country_variable' == "Uae"
298 replace `country_variable' = "Uruguay" if `country_variable' == "Uruguay *"
299 replace `country_variable' = "United States" if `country_variable' == "USA"
300 replace `country_variable' = "United States" if `country_variable' == "Usa"
301 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela *"
302 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela, Rb"
303 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "West Bank Gaza"
304 replace `country_variable' = "West Bank and Gaza" if `country_variable' == "West Bank And
Gaza"
305 replace `country_variable' = "Congo, Rep." if `country_variable' == "Republic of Congo"
306 replace `country_variable' = "North Macedonia" if `country_variable' == "FYR Macedonia"
307 replace `country_variable' = "Hong Kong SAR, China" if `country_variable' == "Hong Kong SAR"
308 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao P.D.R."
309 replace `country_variable' = "Taiwan" if `country_variable' == "Taiwan Province of China"
310 replace `country_variable' = "Bahamas, The" if `country_variable' == "The Bahamas"
311 replace `country_variable' = "Macao SAR, China" if `country_variable' == "Macao SAR"
312 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "Côte d'Ivoire"
313 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "CÔTe D'Ivoire"
314 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo-Brazzaville"
315 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "CÃfÃ'te d'Ivoire"
316 replace `country_variable' = "Cote d'Ivoire" if `country_variable' == "CÃ'te d'Ivoire"
317 replace `country_variable' = "Curaçao" if `country_variable' == "CuraÃsao"
318 replace `country_variable' = "Afghanistan" if `country_variable' == "Afganistan"

```

```

319 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo Dem. Rep"
320 replace `country_variable' = "Somalia" if `country_variable' == "Somalia "
321 replace `country_variable' = "Zimbabwe" if `country_variable' == "Zimbabwe "
322 replace `country_variable' = "Egypt, Arab Rep." if `country_variable' == "Egypt, Arab Rep."
323 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Dem. Rep. "
324 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Dem. Rep"
325 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Democratic
Republic of Congo"
326 replace `country_variable' = "Ghana" if `country_variable' == "Ghana "
327 replace `country_variable' = "Burundi" if `country_variable' == "Burndi"
328 replace `country_variable' = "Philippines" if `country_variable' == "Phillippines"
329 replace `country_variable' = "Bolivia" if `country_variable' == "Bolivua"
330 replace `country_variable' = "Congo, Rep." if `country_variable' == "Republic of the Congo "
331 replace `country_variable' = "Egypt, Arab Rep." if `country_variable' == "Egypt, Arab Rep"
332 replace `country_variable' = "Zambia" if `country_variable' == "Zambia "
333 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' ==
"Micronesia, Fed. Sts."
334 replace `country_variable' = "Yemen, Rep." if `country_variable' == "Yemen Rep."
335 replace `country_variable' = "Lao PDR" if `country_variable' == "Lao PDR "
336 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' == "Korea,
Dem. Rep"
337 replace `country_variable' = "Ethiopia" if `country_variable' == "Ethopia"
338 replace `country_variable' = "Korea, Dem. People's Rep." if `country_variable' ==
"Korea, Dem. Rep"
339 replace `country_variable' = "Micronesia, Fed. Sts." if `country_variable' == "Mirconesia,
Fed. Sts"
340 replace `country_variable' = "Senegal" if `country_variable' == "Sengal"
341 replace `country_variable' = "Serbia & Montenegro" if `country_variable' == "Serbia and
Montenegro"
342 replace `country_variable' = "Tajikistan" if `country_variable' == "Taijikistan"
343 replace `country_variable' = "Yemen, Rep." if `country_variable' == "Yemen, Rep"
344 replace `country_variable' = "Cabo Verde" if `country_variable' == "Cape Verde "
345 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Congo, Dem. Rep."
346 replace `country_variable' = "Congo, Rep." if `country_variable' == "Congo, Rep."
347 replace `country_variable' = "India" if `country_variable' == "India "
348 replace `country_variable' = "Venezuela, RB" if `country_variable' == "Venezuela,
Bolivarian Republic of"
349 replace `country_variable' = "Ethiopia" if `country_variable' == "Ethiopia "
350 replace `country_variable' = "Egypt, Arab Rep." if `country_variable' == "Egypt "
351 replace `country_variable' = "Yemen, Rep." if `country_variable' == "Yemen "
352 replace `country_variable' = "Iraq" if `country_variable' == "Iraq "
353 replace `country_variable' = "Benin" if `country_variable' == "Benin "
354 replace `country_variable' = "Syrian Arab Republic" if `country_variable' == "Syria "
355 replace `country_variable' = "Sri Lanka" if `country_variable' == "SriLanka"
356 replace `country_variable' = "Malawi" if `country_variable' == "Malawi "
357 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' ==
"DemocraticRepublicCongo"
358 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "DRC"
359 replace `country_variable' = "Pakistan" if `country_variable' == "Pakistan "
360 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "DR Congo"
361 replace `country_variable' = "Ukraine" if `country_variable' == "Ukraine "
362 replace `country_variable' = "Congo, Dem. Rep." if `country_variable' == "Democratic
Republic of the Congo (DRC)"
363 replace `country_variable' = "Mali" if `country_variable' == "Mali "
364 replace `country_variable' = "Congo, Rep." if `country_variable' == "Republic of the Congo"
365 replace `country_variable' = "Philippines" if `country_variable' == "Phillipines"
366 replace `country_variable' = "North Macedonia" if `country_variable' == "The Republic of
North Macedonia"
367 replace `country_variable' = "Sao Tome and Principe" if `country_variable' == "Sao Tomé and
Principe"
368 end
369
370
371 **PercentRankInc
372 *****
373 //The following program mimics Excel's PercentRankInc function
374 //For indicators with a static cutoff (Civil Liberties, Political Rights,
375 //Immunization Rates for the higher scorecard pool and Inflation) this is done normally.
376 //However, for indicators with a median based cutoff, countries at the median are set to
377 //the 50th percentile and countries at the minimum are set to the 0th percentile

```

```

378
379 capture program drop PercentRankInc
380 program PercentRankInc
381 args vartest
382
383 generate over = 300 //Dummy value for now, this will be the number of countries scoring
greater than or equal to a given country
384 generate under = 300 //Dummy value for now, this will be the number of countries scoring
less than a given country
385 global count = _N //Number of observations
386 global vartest = "`vartest'"
387
388 forvalues line_change = 1(1)$count { //This loops once for each country.
389     global overcount = -1 //This counts the number of countries over the country in
line_change. It starts at -1 as we do not count the country itself as being greater than
or equal to itself.
390     forvalues line_test = 1(1)$count { //This loops over all the countries again,
comparing them each to the selected country from line_change
391         if `=${vartest}[\`line_change']' <= `=${vartest}[\`line_test']' &
`=${vartest}[\`line_test']' != . { //This if statement augments the overcount whenever a
country from line_test is greater than or equal to the selected country from line_change
392             global overcount = ${overcount} + 1
393         }
394         replace over = ${overcount} in `line_change' //This replaces the variable over with the
overcount macro
395         replace over = . in `line_change' if missing(`=${vartest}[\`line_change']') //This sets
over to missing if the original value is missing.
396     }
397 }
398
399 forvalues line_change = 1(1)$count { //This loops once for each country.
400     global undercount = 0 //This counts the number of countries over the country in
line_change. It starts at 0 since we are not testing less than or equal to, so we will not
count the country against itself.
401     forvalues line_test = 1(1)$count { //This loops over all the countries again, comparing
them each to the selected country from line_change
402         if `=${vartest}[\`line_change']' > `=${vartest}[\`line_test']' &
`=${vartest}[\`line_test']' != . { //This if statement augments the undercount whenever a
country from line_test is less than the selected country from line_change
403             global undercount = ${undercount} + 1
404         }
405         replace under = ${undercount} in `line_change' //This replaces the variable under with
the undercount macro
406         replace under = . in `line_change' if missing(`=${vartest}[\`line_change']') //This sets
over to missing if the original value is missing.
407     }
408 }
409 gen percentile = under/(over+under) //This calculates percentiles for all observations
using the under and over variables.
410
411 //The if statement below forces countries at the median to the 50th percentile, at the max
to the 100th percentile and at the min to the 0th percentile
412 //For all the median based indicators (see above for more information).
413 if "`vartest'" != "Civil_Liberties" & "`vartest'" != "Political_Rights" & "`vartest'" !=
"Inflation" & "`vartest'" != "Immunization_Rate_HISP" {
414     egen median = median(`vartest')
415     egen min = min(`vartest')
416     egen max = max(`vartest')
417     replace percentile = 0.5 if median == `vartest'
418     //replace percentile = 1 if max == `vartest' //In FY21 MCC stopped forcing countrreis tied
for the max to the 100th percentile
419     replace percentile = 0 if min == `vartest'
420 }
421 gen Score_In_FY21 = `vartest'
422 end
423
424 *****
425 ** Import/Clean Raw Data **
426 *****
427 ** The following sections take the necessary data files for MCC's Scorecard,

```



```

428  ** import them, clean them and save the necessary files.  This section
429  ** includes instructions for downloading these files as well as processing them.
430  ** in comments before each section.
431
432  ////////////
433  //  GNI  //
434  ////////////
435  //Countries are divided into two scorecard pools based on GNI per capita.
436  //The following is the construction of that measure from World Bank data.
437
438  //Download Instructions:
439  /*
440  Download the xls file from:
441  https://datacatalog.worldbank.org/dataset/gni-capita-ranking-atlas-method-and-ppp-based
442  Check if columns and sheet names still match
443  Check if the footnotes match the income categories (When there is no GNI data
444  available for a country, in some cases the World Bank still provides a determination
445  of which income group that country would be in, there are captured in footnotes in the
  GNIPC file.)
446
447  Save in the ${raw_data} folder as GNIPC.xls
448
449  Note that MCC uses the July release.  Sometimes the World Bank will change data
450  in between the July and the September release.  The version from the July release
451  is available on MCC's Open Data Portal
452  */
453
454  import excel "${raw_data}\GNIPC July 1 2020 FINAL.xls", sheet("GNIPC") clear
455  //According to the World Bank, these countries fit MCC's definition of LISP and HISP
456  //Therefore the footnotes are translated into income categories
457  //D is country name, E is GNI, F is a footnote for countries missing GNI
458  //Sometimes the world bank will change which footnotes correspond to which income
  categories, check this carefully.
459  //LISP stands for Lower Income Scorecard Pool
460  //HISP stands for Higher Income Scorecard Pool
461  //These are divided by the Historic IDA threshold.
462
463  replace E = "LISP" if F == "k"
464  //replace E = "HISP" if F == "l"
465  replace E = "UMIC" if F == "i"
466  replace E = "HIC" if F == "j"
467  keep D E
468  // The following are not listed as "Independent States" by the State Department,
469  // and therefore do not receive scorecards.  In the following sections, we limit the
470  // dataset by only keeping countries that match a country in this dataset.
471
472  drop if D == "" | D == "Economy" | D == "World" | D == "East Asia & Pacific" /*
473  */ | D == "Europe & Central Asia" | D == "Latin America & Caribbean" /*
474  */ | D == "Middle East & North Africa" | D == "North America" | D == "South Asia" /*
475  */ | D == "Sub-Saharan Africa" | D == "Low income" | D == "Lower middle income" /*
476  */ | D == "Upper middle income" | D == "High income" | D == "West Bank and Gaza"
477
478  //These are placeholder values over the threshold for countries without data.
479  replace E = "15005" if E == "HIC"
480  replace E = "6005" if E == "UMIC"
481  replace E = "105" if E == "LISP"
482
483  rename D Country
484  rename E GNI
485
486  //Now that all of the values for GNI are numeric, we can create a string variable "Income"
  Sorting them by threshold
487  //These threshold categories can be found in the following file labeled "Threshold"
488  //They can be different every year, and so will need to be changed for future iterations
489  destring, replace
490  generate Income = "LISP"
491  replace Income = "HISP" if GNI > 1945
492  replace Income = "UMIC" if GNI > 4045
493  replace Income = "HIC" if GNI > 12535
494

```

```

495 //This program standardizes all of the country names, so that they can be matched with
other data files
496 WBNaming Country
497
498 save "${created_data}\GNI_FY21.dta", replace
499 export excel using "${final_data}\FY21_GNI.xlsx", firstrow(variables) replace
500
501 ////////////////////////////////////////////////////
502 // Thresholds //
503 ////////////////////////////////////////////////////
504 //Countries are divided into two scorecard income pools based on GNI per capita.
505 //The cutoff for these pools is the Historic IDA threshold from the World Bank
506 //The following section demonstrates how these are constructed from the World Bank Data.
507 //This is where the cutoffs for the Income variable in the previous section come from.
508
509 //Download Instructions:
510 /*
511 Download the historical classification by income in XLS format from
512 https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lendi
ng-groups
513 Check if columns and sheet names still match
514 Save in the ${raw_data} folder as OGHIST.xls
515 */
516 import excel "${raw_data}\OGHIST.xls", sheet("Thresholds") clear
517 keep A S-AV
518 keep if A == "Date:" | A == "Bank's fiscal year:" | A == "II. IDA Eligibility /2 /7" | A ==
"Lower middle income" | A == "Upper middle income"
519 sort A
520 replace A = "Threshold" if A == "Bank's fiscal year:"
521 //You may need to install renvars. Enter "help renvars" and install the update.
522 //This command renames the variables with the first row of observations.
523 renvars, map (word(@[1],1))
524 drop in 1
525 foreach var of varlist _all {
526 replace `var' = strreverse(substr(strreverse(`var'),1,4)) in 1 //Takes the year (last 4
digits) only from each value in the first row
527 replace `var' = substr(`var',(strpos(`var'[2],"=")+2),.) in 2 //Isolates the cutoff value
for the second row (the cutoff between the Lower Income Scorecard Pool and the Higher
Income Scorecard Pool)
528 replace `var' = substr(`var',(strpos(`var'[3],"-")+1),.) in 3 //Isolates the cutoff value
for the third row (the cutoff between LMICs and UMICs)
529 replace `var' = substr(`var',(strpos(`var'[4],"-")+1),.) in 4 //Isolates the cutoff value
for the fourth row (the cutoff between UMICs and HICs)
530 replace `var' = substr(`var',"",",",.) //Removes commas from all numbers so they can be
converted to numeric values.
531 }
532 destring, replace
533
534 xpose, clear varname //Reshapes data
535 label variable v1 "Year"
536 label variable v2 "LISP"
537 label variable v3 "HISP"
538 label variable v4 "UMIC"
539 label variable _varname "Fiscal Year"
540 drop in 1
541 order _varname
542 save "${created_data}\Threshold_FY21.dta", replace
543 export excel using "${final_data}\FY21_Threshold.xlsx", firstrow(varlabels) replace
544
545 ////////////////////////////////////////////////////
546 // Population //
547 ////////////////////////////////////////////////////
548 //Population is not used to determine scorecard passage.
549 //However, it is included on the scorecard as a descriptive statistic in the upper left
corner,
550 // so the method for constructing it is listed below
551
552 //Download Instructions:
553 /*
554 Go to https://data.worldbank.org/indicator/SP.POP.TOTL

```

```

555 Download the excel file
556 Save it as World Bank Population Data.xls in the raw data folder
557 */
558 import excel "${raw_data}\World Bank Population Data.xls", sheet("Data") clear
559 //Dropping unnecessary observations with datasource information
560 drop in 3
561 drop in 2
562 drop in 1
563 //Formatting the first row of variables to use renvars
564 foreach var of varlist _all{
565 label var `var' "`='var'[1]'"
566 replace `var' = "Y" + `var'[1] in 1
567 }
568 foreach var of varlist _all{
569 replace `var' = substr("`='var'[1]'", " ", "_",.) in 1
570 }
571 renvars, map (word(@[1],1))
572 drop in 1
573 destring, replace
574 rename YCountry_Name Country
575 drop YCountry_Code YIndicator_Name YIndicator_Code
576 WBNaming Country
577 //This takes out all of the things that are not considered "Independent States" by the
State Department
578 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keepusing() keep(3) //This
removes those observations which were not included in the GNI data above.
579 drop GNI
580 keep Country Y2019
581 export excel using "${final_data}\FY21_Population.xlsx", firstrow(varlabels) replace
582
583
584 ////////////////////////////////////////////////////////////////////
585 // Democratic Rights Indicators (FH) //
586 ////////////////////////////////////////////////////////////////////
587 //The Democratic Rights Indicators are Political Rights and Civil Liberties
588 //They are both sourced from Freedom House's Freedom In The World report.
589 //Download Instructions
590 /*
591 Download the Aggregate Category and Subcategory Scores, 2003-2019 (Excel) file from
592 https://freedomhouse.org/report-types/freedom-world
593 Check if columns and sheet names still match the lines below (i.e. is the sheet still
594 named "FIW06-19" and the variable for country/territory still named CT)
595 Save in the ${raw_data} folder as 2020_Aggregate Category and Subcategory Scores
FIW2003-2020.xlsx
596 */
597 set more off
598 import excel "${raw_data}\2020_Aggregate_Category_and_Subcategory_Scores_FIW_2003-2020.xlsx"
, sheet("FIW06-20") firstrow clear
599 keep if CT == "c" //Only keeping countries
600 keep if Edition == 2020 //Only keeping current year
601 keep CountryTerritory PR CL //Only keeping country name and relevant data values
(PR=Political Rights, CL=Civil Liberties)
602 rename CountryTerritory Country
603
604 **To Merge with GNI data
605 *****
606 WBNaming Country //Renaming countries to match GNI data
607 save "${created_data}\FH_FY21.dta", replace
608
609 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3) //This only keeps
countries included in the GNI data
610
611 keep Country Income CL PR
612 rename CL Civil_Liberties
613 rename PR Political_Rights
614 save "${created_data}\FH Full.dta", replace
615
616 //Percentile are not used for determining passage on this indicator, but they are included
on the scorecard
617 //The following computes the percentile ranks for the Democratic Rights Indicators

```

```

618 //It is worthy of note that MCC's general rule of forcing countries at the median to the
619 50th percentile
620 // does not apply here since percentiles are not used in the calculation of passage
621 global FH_vars "Civil_Liberties Political_Rights"
622 foreach var of global FH_vars {
623 //Percentile Ranks for LISPs
624 preserve
625 keep if Income == "LISP"
626 PercentRankInc `var' //This program is listed above, and calculates the percentile rank
627 using MCC's Methodology
628 save "${created_data}\FH `var' LISP.dta", replace
629 restore
630 //Percentile Ranks for HISPs
631 preserve
632 keep if Income == "HISP"
633 PercentRankInc `var'
634
635 //Appending other scorecard pool data
636 append using "${created_data}\FH `var' LISP.dta"
637 keep Country percentile Score_In_FY21 Income
638 gen Indicator = "`var'"
639 rename percentile Percentile
640 if "`var'" == "Civil_Liberties" { //We need to use an if statement here because we can't
641 put a local macro at the beginning of the filename
642 save "${created_data}\Civil Liberties.dta", replace
643 }
644 else {
645 save "${created_data}\Political Rights.dta", replace
646 }
647 export excel "${final_data}\FY21_`var'.xlsx", firstrow(variables) replace
648 restore
649 }
650
651
652 ///////////////////////////////////////////////////
653 // Education Indicators //
654 ///////////////////////////////////////////////////
655
656 //Download Instructions:
657 /*
658 Go to http://data.uis.unesco.org/
659 On the left panel select National Monitoring > National Monitoring > National Monitoring
660 Full Dataset
661 Click Customise > Selection > Indicator
662 Click Unselect Items > Unselect all then Collapse all
663 Click National > Indicator > Government expenditure on education as a percentage of GDP
664 > Government expenditure on primary education as a percentage of GDP (%) (XGDP_1_FSGOV)
665
666 Then select Participation > Enrollment Ratios >
667 National > Indicator> Number of students and enrolment rates by level of education
668 > Enrolment ratios > Gross enrolment ratio by level of education > Gross enrolment ratio,
669 lower secondary, female (%) (GER_2_F)
670
671 Click Time > Select latest data > Select last 50 years
672
673 Click View Data
674 Go to Export > Text File (CSV) > Download
675
676 Check if columns and sheet names still match
677 Save in the ${raw_data} folder as Ed Nat Data YEAR.csv
678
679 Completion is now with the SDG indicator:
680 Click Customise > Selection > Indicator
681 Click Unselect Items > Unselect all then Collapse all
682 Then select SDG > Sustainable Development Goal 4 >
683 Target 4.1: By 2030 ensure that all girls and boys complete free, equitable, and quality

```

```

683 primary and secondary education leading to relevant and effective learning outcomes
684 > 4.1.3 Gross intake ratio to the last grade (primary education, lower secondary education) >
685 Gross intake ratio to the last grade of primary education, female (%)
686 Click Time > Select latest data > Select last 50 years
687
688 Click View Data
689 Go to Export > Text File (CSV) > Download
690
691 Check if columns and sheet names still match
692 Save in the ${raw_data} folder as Ed SDG Data YEAR.csv
693 */
694 set more off
695 import delimited "${raw_data}\Ed Nat Data 2020.csv", clear
696 rename inatmon_ind indcode
697 save "${created_data}\Ed Nat Data 2020.dta", replace
698 import delimited "${raw_data}\Ed SDG Data 2020.csv", clear
699 rename isdg_ind indcode
700 append using "${created_data}\Ed Nat Data 2020.dta"
701
702 drop v6
703 //These countries would be removed when the GNI data is merged anyway, but we need to drop
704 them early due to the way the reshaping process handles duplicates
705 drop if country == "World" | country == "Latin America and the Caribbean" | country ==
706 "Small Island Developing States" /*
707 */ | country == "Africa (Northern)" | country == "Africa (Sub-Saharan) " | country == "Arab
708 States" /*
709 */ | country == "Arab States" | country == "Asia (Central and Southern)" | country == "Asia
710 (Central)" /*
711 */ | country == "Asia (Eastern and South-eastern)" | country == "Asia (Eastern)" | country
712 == "Asia (South-eastern)" /*
713 */ | country == "Asia (Southern)" | country == "Asia (Western)" | country == "Africa
714 (Sub-Saharan)" | country == "East Asia and the Pacific" /*
715 */ | country == "Central Asia" | country == "Central and Eastern Europe" | country == "High
716 income countries" | country == "Landlocked Developing Countries" /*
717 */ | country == "Least Developed Countries" | country == "Low income countries" | country ==
718 "Lower middle income countries" | country == "Middle income countries" /*
719 */ | country == "North America and Western Europe" | country == "Northern America" | country
720 == "Northern America and Europe" | country == "Oceania" /*
721 */ | country == "Oceania (Australia/New Zealand)" | country == "Oceania (excl.
722 Australia/New Zealand)" | country == "South and West Asia" | country == "Sub-Saharan Africa"
723 /*
724 */ | country == "Upper middle income countries" | country == "Western Asia and Northern
725 Africa" | country == "Europe" | country == "Northern America" | country == "Northern
726 America "
727
728 //Renaming countries for consistency
729 WBNaming country
730 rename time year
731
732 ////Reshaping the data into a "wide" format so it is easier to use
733 **EdExpense
734 preserve
735 keep if indicator == "Government expenditure on primary education as a percentage of GDP (%)"
736 keep country year value
737 reshape wide value, i(country) j(year)
738 save "${created_data}\EdExpense FY21.dta" , replace
739 restore
740
741 **Primary
742 preserve
743 keep if indicator == "Gross intake ratio to the last grade of primary education, female (%)"
744 keep country year value
745 reshape wide value, i(country) j(year)
746 save "${created_data}\Primary FY21.dta" , replace
747 restore
748
749 **Secondary

```

```

739 keep if indicator == "Gross enrolment ratio, lower secondary, female (%)"
740 keep country year value
741 reshape wide value, i(country) j(year)
742 save "${created_data}\Secondary FY21.dta" , replace
743
744
745 **Merging
746 *****
747 //For these indicators, MCC takes the most recent data point from the last 7 years of data
748 (including the current year, 2019)
749 //Other documentation states that we take the last 6 years of data, this is because only
750 one country has data for 2019
751 //Therefore it is the 6 most recent years of data starting in 2018 or the 7 most recent
752 starting in 2019
753 //The following code takes the most recent value for each indicator in the last 7 years
754 (since 2013)
755 set more off
756 //Ed Expenditures
757 use "${created_data}\EdExpense FY21.dta", clear
758 keep country value2014-value2020
759 egen EdExpense = rowlast(value*) //Using the last (most recent) value, note that the
760 variables must be sorted chronologically for this to work
761 keep country EdExpense
762 //Primary Completion
763 merge 1:1 country using "${created_data}\Primary FY21.dta" , nogen
764 keep country EdExpense value2014-value2020 //Limiting to the last 7 years
765 egen Primary = rowlast(value*) //Taking the most recent data
766 keep country Primary EdExpense
767 //Secondary Enrollment
768 merge 1:1 country using "${created_data}\Secondary FY21.dta", nogen
769 replace country = "Cote d'Ivoire" if country == "CÃ´te d'Ivoire"
770 replace country = "Cote d'Ivoire" if country == "CÃ´te d'Ivoire"
771 keep country Primary EdExpense value2014-value2020
772 egen Secondary = rowlast(value*)
773 keep country Secondary EdExpense Primary
774 rename country Country
775 WBNaming Country
776 merge 1:1 Country using "${created_data}\GNI_FY21.dta" , nogen keep(3)
777
778 global EdIndicators "Secondary Primary EdExpense"
779 save "${created_data}\Ed Full.dta", replace
780
781 //The following sections calculate percentile ranks for each of the three education
782 indicators and generate the files:
783 preserve
784 keep if Income == "LISP"
785 PercentRankInc EdExpense
786 save "${created_data}\Ed EdExpense LISP.dta", replace
787 restore
788 preserve
789 keep if Income == "HISP"
790 PercentRankInc EdExpense
791 append using "${created_data}\Ed EdExpense LISP.dta"
792 keep Country Income percentile Score_In_FY21
793 gen Indicator = "Primary_Education_Expenditures"
794 rename percentile Percentile
795 save "${created_data}\Primary Education Expenditures.dta", replace
796 export excel using "${final_data}\FY21_Primary_Education_Expenditures.xlsx", firstrow(
797 variables) replace
798 restore
799
800 preserve
801 keep if Income == "LISP"
802 PercentRankInc Primary
803 keep Country Income percentile Score_In_FY21
804 gen Indicator = "Primary_Girls_Education_Completion"
805 rename percentile Percentile
806 save "${created_data}\Primary Girls Education Completion.dta", replace
807 export excel using "${final_data}\FY21_Primary_Girls_Education_Completion.xlsx", firstrow(
808 variables) replace

```

```

801 restore
802
803 preserve
804 keep if Income == "HISP"
805 PercentRankInc Secondary
806 keep Country Income percentile Score_In_FY21
807 gen Indicator = "Secondary_Girls_Education_Enrollment"
808 rename percentile Percentile
809 save "${created_data}\Secondary Girls Education Enrollment.dta", replace
810 export excel using "${final_data}\FY21_Secondary_Girls_Education_Enrollment.xlsx", firstrow(
variables) replace
811 restore
812
813
814 ////////////////////////////////////////////////////
815 // Immunization Rate Indicator //
816 ////////////////////////////////////////////////////
817 //Download Instructions
818 /*
819 Download from https://www.who.int/immunization/monitoring_surveillance/data/en/
820 Under 4. Immunization coverage or administered doses, select:
821 4.6 WHO/UNICEF Estimates of National Immunization Coverage (excel)
822
823 Check if columns and sheet names still match (i.e. is the sheet still called MCV1 and the
Country variable still called Cname)
824 Save in the ${raw_data} folder as WHO coverage_estimates_series.xls
825 */
826 //This indicator is an average of two measures of immunization: MCV1 and DTP3
827 //The following code performs this calculation and outputs the final file.
828 set more off
829 import excel "${raw_data}\WHO coverage_estimates_series.xls", sheet("MCV1") clear
830
831 //MCV1
832 foreach var of varlist E-AR { //This loop allows the first row to be renamed as Y{year}
(e.g. Y2016) since Stata cannot take numbers as variable names
833 replace `var' = "Y" + `var'[1] in 1
834 }
835 renvars, map (word(@[1],1))
836 drop in 1
837 destring, replace
838 rename Cname Country
839 keep Country Y*
840 WBNaming Country
841 order _all, sequential
842 keep Country Y2019
843 save "${created_data}\Immunization MCV FY21.dta" , replace
844
845 //DTP3
846 import excel "${raw_data}\WHO coverage_estimates_series.xls", sheet("DTP3") clear
847 foreach var of varlist E-AR { //This loop allows the first row to be renamed as Y{year}
(e.g. Y2016) since Stata cannot take numbers as variable names
848 replace `var' = "Y" + `var'[1] in 1
849 }
850 renvars, map (word(@[1],1))
851 drop in 1
852 destring, replace
853 rename Cname Country
854 keep Country Y*
855 WBNaming Country
856 order _all, sequential
857 keep Country Y2019
858 save "${created_data}\Immunization DTP FY21.dta" , replace
859
860 **Merging
861 *****
862 //Merging both files and then merging them with the GNI data
863 set more off
864 use "${created_data}\Immunization MCV FY21.dta", clear
865 keep Country Y2019
866 rename Y2019 MCV1

```

```

867 merge 1:1 Country using "${created_data}\Immunization DTP FY21.dta"
868 keep Country Y2019 MCV1
869 rename Y2019 DTP3
870 WBNaming Country
871 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
872
873 egen Score = rowmean(MCV1 DTP3) //Taking the average of the two values
874
875 save "${created_data}\Imm Full.dta", replace
876
877 preserve
878 keep if Income == "LISP"
879 // Starting in FY21, when the scorecard pool median for immunization rates
880 // data is below 90% the data is treated as a median based indicator (judged only
881 // on if the value is above or below the median, ties for the median are
882 // forced to 50th percentile) by naming it Immunization_Rate_HISP. If the median
883 // for the scorecard income pool is above 90% then the data is treated as an
884 // absolute threshold indicator (judged against the threshold of 90%, ties for
885 // the median are NOT forced to the 50th percentile) by naming it
886 // Immunization_Rate_LISP. The code below implements this change. This naming
887 // structure corresponds to the rules in the PercentRankInc Program above.
888 qui sum Score, d
889 local Imm_Med = `r(p50)'
890 if `Imm_Med'>90 {
891     //PercentRankInc Treats the variables named HISP as absolute and variables
892     //named LISP as median based due to the old rules for this indicator.
893     rename Score Immunization_Rate_HISP
894     PercentRankInc Immunization_Rate_HISP
895     rename Immunization_Rate_HISP Score
896 }
897 else {
898     rename Score Immunization_Rate_LISP
899     PercentRankInc Immunization_Rate_LISP
900     rename Immunization_Rate_LISP Score
901 }
902 }
903 save "${created_data}\Imm LISP.dta", replace
904
905 restore
906 keep if Income == "HISP"
907 if `Imm_Med'>90 {
908     //PercentRankInc Treats the variables named HISP as absolute and variables
909     //named LISP as median based due to the old rules for this indicator.
910     rename Score Immunization_Rate_HISP
911     PercentRankInc Immunization_Rate_HISP
912     rename Immunization_Rate_HISP Score
913 }
914 else {
915     rename Score Immunization_Rate_LISP
916     PercentRankInc Immunization_Rate_LISP
917     rename Immunization_Rate_LISP Score
918 }
919 }
920
921 append using "${created_data}\Imm LISP.dta"
922 keep Country Income percentile Score_In_FY21
923 gen Indicator = "Immunization_Rate"
924 rename percentile Percentile
925 keep if Income == "LISP" | Income == "HISP"
926 save "${created_data}\Immunization Rate.dta", replace
927 export excel using "${final_data}\FY21_Immunization_Rate.xlsx", firstrow(variables) replace
928
929
930 //////////////////////////////////////////////////
931 // Health Expenditures //
932 //////////////////////////////////////////////////
933
934 //Download Instructions
935 /*
936 Go to http://apps.who.int/nha/database/Select/Indicators/en

```



```

937 Select Indicators > Financing Sources >
938 Domestic General Government Health Expenditure (GGHE-D) as % of Gross Domestic Product (GDP)
939 Click Next, select all countries, click next
940 Select all years, click next
941 Select % Gross domestic product (GDP)
942 click view data and build report
943 Click Excel Workbook
944
945 Save as NHA indicators.xlsx in the raw data folder
946 check to see if the columns and rows still match.
947 */
948 set more off
949 import excel "${raw_data}\NHA indicators.xlsx", sheet("Table") clear
950 drop B C
951 drop in 2
952
953 foreach var of varlist D-U { //replacing the first row with values that have characters in
954 them so that variables can be renamed using renvars
955 replace `var' = "Y" + `var'[1] in 1
956 }
957 renvars, map (word(@[1],1))
958 drop in 1
959 rename Countries Country
960 WBNaming Country
961 destring, replace
962 save "${created_data}\Health FY21.dta", replace
963
964 **Merging
965 *****
966 keep Country Y2017 //Keep the most recent year
967 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
968 save "${created_data}\Health Full.dta", replace
969
970 preserve
971 keep if Income == "LISP"
972 PercentRankInc Y2017
973 save "${created_data}\Health LISP.dta", replace
974 restore
975 keep if Income == "HISP"
976 PercentRankInc Y2017
977 save "${created_data}\Health HISP.dta", replace
978 append using "${created_data}\Health LISP.dta"
979
980 keep Country Income percentile Score_In_FY21
981 gen Indicator = "Health_Expenditures"
982 rename percentile Percentile
983 save "${created_data}\Health Expenditures.dta", replace
984 export excel using "${final_data}\FY21_Health_Expenditures.xlsx", firstrow(variables) replace
985
986 ////////////////////////////////////////////////////
987 // Gender in the Economy //
988 ////////////////////////////////////////////////////
989
990
991 //Download Instructions:
992 /*
993
994 In 2020, Women, Business and the Law substantially changed their underlying data. MCC is
995 still able
996 to produce its Gender in the Economy indicator from the available data, however multiple
997 files must be accessed.
998 Additionally, MCC must use the information provided by WBL in the "About" tab of the 50
999 Year Panel
1000 About disaggregated values. This information must be manually coded into the data by MCC.
1001 The three required files can be found here:
1002
1003 50 Year Data File
1004 The Data files used by MCC can be found here:
1005 https://wbl.worldbank.org/content/dam/wbl/data/WBL-Dataset-50years.dta?logActivity=true

```

1002 or here
<https://web.archive.org/web/20200815143655/https://wbl.worldbank.org/content/dam/wbl/data/WBL-Dataset-50years.dta?logActivity=true>

1003 The current version of this data file can be found here under Women, Business and the Law Data for 1971-2021

1004 (Download file (STATA)) <https://wbl.worldbank.org/en/wbl-data>

1005 Name this file "WBL-Dataset-50years.dta" and save in the raw data folder

1006

1007 Additional Data File:

1008 Can be found here
https://wbl.worldbank.org/content/dam/sites/wbl/documents/2021/02/WBLAdditionalData_07July2020.xlsx or

1009 <http://pubdocs.worldbank.org/en/666161589301125247/WBL-Additional-Data.xlsx>

1010 Name this file "WBLAdditionalData.xlsx" and save in the raw data folder

1011

1012 2020 Data File Release

1013 Can be found here:
<http://pubdocs.worldbank.org/en/625831591105047863/WBL-50YearPanelDetails-Web-01Jun2020.xlsx> or

1014 here
<https://web.archive.org/web/20210305161933/http://pubdocs.worldbank.org/en/625831591105047863/WBL-50YearPanelDetails-Web-01Jun2020.xlsx>

1015 Name this file "WBL50YearPanelDetailsWeb01Jun2020.xlsx" and save in the raw data folder

1016

1017 The Numbers below correspond to the following questions:

1018 Accessing Institutions

1019 1. Can an unmarried woman obtain a national ID card in the same way as an unmarried man? (1/2 point)

1020 2. Can a married woman obtain a national ID card in the same way as a married man? (1/2 point)

1021 3. Can an unmarried woman travel outside the country in the same way as an unmarried man? (1/2 point)

1022 4. Can a married woman travel outside the country in the same way as a married man? (1/2 point)

1023 5. Can an unmarried woman travel outside her home in the same way as an unmarried man? (1/2 point)

1024 6. Can a married woman travel outside her home in the same way as a married man? (1/2 point)

1025 7. Can an unmarried woman get a job or pursue a trade or profession in the same way as an unmarried man? (1/2 point)

1026 8. Can a married woman get a job or pursue a trade or profession in the same way as a married man? (1/2 point)

1027 9. Can an unmarried woman sign a contract in the same way as an unmarried man? (1/2 point)

1028 10. Can a married woman sign a contract in the same way as a married man? (1/2 point)

1029 11. Can an unmarried woman register a business in the same way as an unmarried man? (1/2 point)

1030 12. Can a married woman register a business in the same way as a married man? (1/2 point)

1031 13. Can an unmarried woman open a bank account in the same way as an unmarried man? (1/2 point)

1032 14. Can a married woman open a bank account in the same way as a married man? (1/2 point)

1033 15. Can an unmarried woman choose where to live in the same way as an unmarried man? (1/2 point)

1034 16. Can a married woman choose where to live in the same way as a married man? (1/2 point)

1035 17. Can an unmarried woman confer citizenship to children in the same way as an unmarried man? (1/2 point)

1036 18. Can a married woman confer citizenship to children in the same way as a married man? (1/2 point)

1037 19. Can an unmarried woman be "head of household" in the same way as an unmarried man? (1/2 point)

1038 20. Can a married woman be "head of household" in the same way as a married man? (1/2 point)

1039 21. If customary law is recognized as a valid source of law under the constitution, is it invalid if it violates constitutional provisions on nondiscrimination or equality? (1 point)

1040 22. Do married couples jointly share legal responsibility for financially maintaining the family's expenses? (1 point)

1041 Using Property

1042 23. Does the law provide for valuation of nonmonetary contributions? (1 point)

1043 24. Who administers marital property? (1 point for "husband")

1044 25. Do unmarried men and unmarried women have equal ownership rights to property? (1/2 point)

1045 26. Do married men and married women have equal ownership rights to property? (1/2 point)

1046 27. Do sons and daughters have equal rights to inherit assets from their parents? (1 point)

1047 28. Do female and male surviving spouses have equal rights to inherit assets? (1 point)

```

1048 Going to Court
1049 29. Does a woman's testimony carry the same evidentiary weight in court as a man's? (1 point)
1050 Getting a Job
1051 30. Can nonpregnant and nonnursing women work the same night hours as men? (1 point)
1052 31. Can nonpregnant and nonnursing women do the same jobs as men? (1 point)
1053 Protecting Women from Violence
1054 32. Is there domestic violence legislation? (1 point)
1055 33. Are there clear criminal penalties for domestic violence? (1 point)
1056 34. Is there a specialized court or procedure for cases of domestic violence? (1 point)
1057 35. Is there legislation that specifically addresses sexual harassment? (1 point)
1058 36. Are there criminal penalties for sexual harassment in employment? (1 point)
1059 37. What is the legal age of marriage for girls? (1 point for ages < 18 or no data)
1060 38. Are there any exceptions to the legal age of marriage? (1 point for "yes")
1061 39. Does the law prohibit or invalidate child or early marriage? (1 point)
1062 40. Are there penalties in the law for authorizing or knowingly entering into child or
early marriage? (1 point)
1063 */
1064
1065
1066 set more off
1067 use "${raw_data}\WBL-Dataset-50years.dta", clear
1068 keep EconomyCode Incomegroup Year gr1_2trvlctrymrd gr1_3trvlhmmrd gr1_4whlivermrd
gr2_5profhmmrd gr2_7sexhrssemp gr2_8sexcomb gr3_10nprgeqnight gr3_11jobshazard
gr3_12industry gr4_14hhmrd gr4_15domleg gr6_23cntrcthmmrd gr6_24regbusmrd gr6_25bnkaccmrd
gr6_26disgend gr7_27prtyeqownmrd bth gr7_28prtyeqsondght gr7_29prtyeqsuvrspse
gr7_30prtylegadmin gr7_31valnonmnty
1069 keep if Year == 2020 //Changed 9.4.20
1070 rename Code EconomyCode
1071 save "${created_data}\WBL Index Data.dta", replace
1072 //20
1073
1074 import excel "${raw_data}\WBLAdditionalData.xlsx", clear firstrow sheet("2019")
1075 //import excel "${raw_data}\Archive\WBLAdditionalData.xlsx", clear firstrow sheet("2019")
//7.17.20
1076 keep Economy Year EconomyCode IncomeGroup Whatisthelegalageofmarriag
Arethereanyexceptionstothe Ismarriageunderthelegalage Aretherepenaltiesforauthoriz
Istherelegislationthatspecif Doesthelegislationestablishc Isthereaspecializedcourtor
Canawomanobtainanationalid /*Canawomanlegallyconfercitiz*/ Canamarriedwomanlegallyconf
Cananunmarriedwomanlegallyc Domarriedcouplesjointlyshare Doesawomanstestimonycarryt
Iscustomarylawavalidsource Ifcustomarylawisavalidsour //7.17.20
1077 rename IncomeGroup Incomegroup
1078 save "${created_data}\WBL Additional Data.dta", replace
1079 //14
1080
1081 import excel "${raw_data}\WBL50YearPanelDetailsWeb01Jun2020.xlsx", clear firstrow sheet(
"Workplace")
1082
1083 keep EconomyCode EconomyName N
1084 rename N CriminalPenaltiesSH
1085
1086 gen Year = 2019
1087 rename EconomyName Economy
1088 replace Economy = "Puerto Rico (U.S.)" if EconomyCode == "PRI"
1089 drop in 1
1090 merge 1:1 Economy using "${created_data}\WBL Index Data.dta", nogen
1091 merge 1:1 Economy using "${created_data}\WBL Additional Data.dta"
1092
1093 //1// gr7_29prtyeqsuvrspse
1094 //2// gr7_28prtyeqsondght
1095 //3// gr7_31valnonmnty
1096 //4// gr4_15domleg
1097 //5// Istherelegislationthatspecif
1098 //6// gr7_30prtylegadmin
1099 //7// gr6_23cntrcthmmrd
1100 /*8*/ gen nw_contract_unm = "Yes" //Yes is a good thing
1101 //9// gr1_2trvlctrymrd
1102 /*10*/ gen nw_travel_abd_unm = "Yes" //Yes is a good thing
1103 replace nw_travel_abd_unm = "No" if Economy == "Sudan" //No is a bad thing
1104 //11// gr6_25bnkaccmrd
1105 /*12*/ gen nw_bank account_unm = "Yes" //Yes is a good thing

```

```

1106 //13// gr6_24regbusmrd
1107 /*14*/ gen nw_start_business_unm = "Yes" //Yes is a good thing
1108 //15// gr1_3trvlhmmrd
1109 /*16*/ gen nw_travel_home_unm = "Yes" //Yes is a good thing
1110 //17// Isthereaspecializedcourtor
1111 //18// Doesthelegislationestablishc
1112 //19// gr2_5profhmmrd
1113 /*20*/ gen nw_get_job_unm = "Yes" //Yes is good
1114 //21// gr7_27prtyeqownmrdboth // FIXED 7.15.20 // This does not track. There are many more
countries listed with restrictions here than listed in Tea's email
1115 /*22*/ gen nw_property_own_unm = "Yes" //Yes is good
1116     replace nw_property_own_unm = "No" if Economy == "Tonga" //| Economy == "Eswatini"
//No is bad //FIXED 7.15.20//This does not track. See above.
1117 //23// gr1_4whlivermrd
1118 /*24*/ gen nw_where_live_unm = "Yes"
1119 //25// gr4_14hhmrd
1120 /*26*/ gen nw_HoH_unm = "Yes" //Yes is a good thing
1121     replace nw_HoH_unm = "No" if Economy == "Sudan" //No is a bad thing
1122 //27// gr3_10nprgeqnight
1123 /*28*/ gen nw_job_type = (gr3_11jobshazard & gr3_12industry)
tostring nw_job_type, force replace
1124     replace nw_job_type = "Yes" if nw_job_type == "1"
1125     replace nw_job_type = "No" if nw_job_type == "0"
1126 //29// CriminalPenaltiesSH
1127 //30// Canawomanobtainanationalid
1128 /*31*/ gen nw_nat_id_unm = "Yes" //Yes is good
1129     replace nw_nat_id_unm = "No" if Economy == "Oman"
1130 //32// Arethereanyexceptionstothe
1131 //33// Aretherepenaltiesforauthoriz
1132 /*34*/ gen nw_citizen_mar = Canamarriedwomanlegallyconf
1133 /*35*/ gen nw_citizen_unm = Cananunmarriedwomanlegallyc
1134 //36// Domarriedcouplesjointlyshare
1135 //37// Doesawomanstestimonycarryt
1136 //38// Ismarriageunderthelegalage
1137 //39// Ifcustomarylavisavalidsour Iscustomarylavalidsource
1138 //40// Whatisthelegalageofmarriag
1140
1141 keep Economy Year gr7_29prtyeqsuvrspse gr7_28prtyeqsondght gr7_31valnonmnty gr4_15domleg
Istherelegislationthatspecif gr7_30prtylegadmin /* //6
1142 */ Isthereaspecializedcourtor Doesthelegislationestablishc gr3_10nprgeqnight nw_job_type
CriminalPenaltiesSH Arethereanyexceptionstothe /* //6
1143 */ Aretherepenaltiesforauthoriz Domarriedcouplesjointlyshare Doesawomanstestimonycarryt
Ismarriageunderthelegalage /* //4
1144 */ Ifcustomarylavisavalidsour Iscustomarylavalidsource Whatisthelegalageofmarriag /* //2
1145 */ gr6_23cntrcthmrd nw_contract_unm gr1_2trvlctrymrd nw_travel_abd_unm gr6_25bnkaccmrd
nw_bank_account_unm gr6_24regbusmrd /* //7
1146 */ nw_start_business_unm gr1_3trvlhmmrd nw_travel_home_unm gr2_5profhmmrd nw_get_job_unm
gr7_27prtyeqownmrdboth /* //6
1147 */ nw_property_own_unm gr1_4whlivermrd nw_where_live_unm gr4_14hhmrd nw_HoH_unm
Canawomanobtainanationalid nw_nat_id_unm /* //7
1148 */ nw_citizen_mar nw_citizen_unm //2
1149
1150
1151 global coded "gr1_2trvlctrymrd gr1_3trvlhmmrd gr1_4whlivermrd gr2_5profhmmrd
gr3_10nprgeqnight gr4_14hhmrd gr4_15domleg gr6_23cntrcthmrd gr6_24regbusmrd
gr6_25bnkaccmrd gr7_27prtyeqownmrdboth gr7_28prtyeqsondght gr7_29prtyeqsuvrspse
gr7_30prtylegadmin gr7_31valnonmnty"

1152
1153 foreach var of global coded {
1154     decode `var' , gen(`var'_d)
1155     drop `var'
1156     rename `var'_d `var'
1157 }
1158 //Questions worth 1 point
1159 global fullpoint_indicators "gr7_29prtyeqsuvrspse gr7_28prtyeqsondght gr7_31valnonmnty
gr4_15domleg Istherelegislationthatspecif gr7_30prtylegadmin Isthereaspecializedcourtor
Doesthelegislationestablishc gr3_10nprgeqnight nw_job_type CriminalPenaltiesSH
Aretherepenaltiesforauthoriz Domarriedcouplesjointlyshare Doesawomanstestimonycarryt
Ismarriageunderthelegalage Ifcustomarylavisavalidsour"
1160

```

```

1161 //Questions worth half a point
1162 global halfpoint_indicators "gr6_23cntrcrthmrd nw_contract_unm gr1_2trvlctrymrd
nw_travel_abd_unm gr6_25bnkaccmrd nw_bank_account_unm gr6_24regbusmrd nw_start_business_unm
gr1_3trvlhmrd nw_travel_home_unm gr2_5profhmmrd nw_get_job_unm gr7_27prtyeqownmrdboth
nw_property_own_unm gr1_4whlivermrd nw_where_live_unm gr4_14hhmrd nw_HoH_unm
Canawomanobtainanationalid nw_nat_id_unm nw_citizen_mar nw_citizen_unm"

1163
1164 //Assigning one point to the full point indicators
1165 foreach fullpoint of global fullpoint_indicators {
1166 replace `fullpoint' = "1" if `fullpoint' == "No"
1167 replace `fullpoint' = "0" if `fullpoint' == "Yes"
1168 replace `fullpoint' = "0" if `fullpoint' == "yes"
1169 replace `fullpoint' = "0" if `fullpoint' == "Yes "
1170 replace `fullpoint' = "1" if `fullpoint' == "Husband"
1171 replace `fullpoint' = "1" if `fullpoint' == "husband"
1172 replace `fullpoint' = "0" if `fullpoint' == "Other"
1173 replace `fullpoint' = "0" if `fullpoint' == "other"
1174 replace `fullpoint' = "0" if `fullpoint' == "Both must agree"
1175 replace `fullpoint' = "0" if `fullpoint' == "both must agree"
1176 replace `fullpoint' = "0" if `fullpoint' == "Original owner"
1177 replace `fullpoint' = "0" if `fullpoint' == "original owner"
1178 replace `fullpoint' = "0" if `fullpoint' == "Original Owner"
1179 replace `fullpoint' = "0" if `fullpoint' == "Separate with spousal consent"
1180 replace `fullpoint' = "0" if `fullpoint' == "separate with spousal consent"
1181 replace `fullpoint' = "." if `fullpoint' == "N/A"
1182 replace `fullpoint' = "1" if `fullpoint' == ".."
1183 }
1184 //Assigning a half point to the half point indicators
1185 foreach halfpoint of global halfpoint_indicators {
1186 replace `halfpoint' = "0.5" if `halfpoint' == "No"
1187 replace `halfpoint' = "0" if `halfpoint' == "Yes"
1188 replace `halfpoint' = "0" if `halfpoint' == "yes"
1189 replace `halfpoint' = "." if `halfpoint' == "N/A"
1190 replace `halfpoint' = "0.5" if `halfpoint' == ".."
1191 }
1192
1193 //Assigning the correct point value to the remaining questions on child marriage and
customary law
1194 rename Arethereanyexceptionstothe any_exceptions_to_the_leg_agef
1195
1196 replace any_exceptions_to_the_leg_agef = "1" if any_exceptions_to_the_leg_agef == "Yes"
1197 replace any_exceptions_to_the_leg_agef = "1" if any_exceptions_to_the_leg_agef == "yes"
1198 replace any_exceptions_to_the_leg_agef = "0" if any_exceptions_to_the_leg_agef == "No"
1199 replace any_exceptions_to_the_leg_agef = "0" if any_exceptions_to_the_leg_agef == "no"
1200 replace any_exceptions_to_the_leg_agef = "1" if any_exceptions_to_the_leg_agef == ".."
1201
1202 rename Whatisthelegalageofmarriag leg_age_of_marr_girls
1203 replace leg_age_of_marr_girls = "1" if leg_age_of_marr_girls == ".."
1204 destring, replace
1205
1206 replace leg_age_of_marr_girls = 1 if leg_age_of_marr_girls < 18
1207 replace leg_age_of_marr_girls = 0 if leg_age_of_marr_girls >= 18
1208 drop Year Iscustomarylavalidsource
1209 unab Allvars : _all
1210 unab exclude : Economy
1211 local Addlist : list Allvars - exclude
1212 di "`Addlist'"
1213 preserve
1214 collapse (sum) `Addlist'
1215 restore
1216 egen Gender_score = rowtotal(`Addlist')
1217 rename Economy Country
1218 keep Country Gender_score
1219
1220 label variable Gender_score "2019"
1221 WBNaming Country
1222 save "${created_data}\Gender FY21.dta", replace
1223
1224 **Merging
1225 *****

```

```

1226
1227 use "${created_data}\Gender FY21.dta", clear
1228 keep Country Gender_score
1229 WBNaming Country
1230 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1231
1232 save "${created_data}\Gender Full.dta", replace
1233 preserve
1234 keep if Income == "LISP"
1235 PercentRankInc Gender_score
1236 replace percentile = 1-percentile //Note this is different from other indicators due to the
inverse scale
1237 save "${created_data}\Gender LISP.dta", replace
1238 restore
1239 keep if Income == "HISP"
1240 PercentRankInc Gender_score
1241 replace percentile = 1-percentile //Note this is different from other indicators due to the
inverse scale
1242 save "${created_data}\Gender HISP.dta", replace
1243 append using "${created_data}\Gender LISP.dta"
1244
1245
1246 keep Country Income percentile Score_In_FY21
1247 gen Indicator = "Gender_In_The_Economy"
1248 rename percentile Percentile
1249 save "${created_data}\Gender In The Economy.dta", replace
1250 export excel using "${final_data}\FY21_Gender_In_The_Economy.xlsx", firstrow(variables)
replace

1251
1252 ////////////////////////////////////////////////////
1253 // Natural Resource Protection //
1254 ////////////////////////////////////////////////////
1255
1256 //Download Instructions:
1257 /*
1258 Go to CIESIN's Natural Resource Management Index
1259 http://www.ciesin.columbia.edu/data/nrpi-chi-2020/
1259 If the link does not redirect you to SEDAC's webpage, select "NRPI and CHI data download".
If it does redirect:
1260 Click on Data Sets, and select the most recent release of the Natural resource protection
and Child Health Indicators
1261 Click on Data Download. You may need to log into NASA's website to view the data.
1262 Select the zip release
1263
1264 Note, some of this data has been historically revised. MCC is working with CIESIN to make
sure the original
1265 data that was shared on the scorecards is published.
1266
1267 Save as the file referenced below in the raw data folder
1268 */
1269
1270
1271 import excel "${raw_data}\NRPI_CHI.xlsx", sheet("NRPI_2020") firstrow clear
1272 drop if NRPI_v2020_20 == "NA" //Duplicate Sudan
1273 destring _all, replace
1274 WBNaming CountryName
1275 label var CountryName "Country"
1276 keep CountryName NRPI*
1277 save "${created_data}\NRP FY21.dta", replace
1278
1279 **Merging
1280 *****
1281
1282 use "${created_data}\NRP FY21.dta", clear
1283 rename CountryName Country
1284 keep Country NRPI_v2020_20
1285 WBNaming Country
1286 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1287 save "${created_data}\NRP Full.dta", replace
1288 preserve

```

```

1289 keep if Income == "LISP"
1290 PercentRankInc NRPI_v2020_20
1291 save "${created_data}\NRP LISP.dta", replace
1292 restore
1293 keep if Income == "HISP"
1294 PercentRankInc NRPI_v2020_20
1295 save "${created_data}\NRP HISP.dta", replace
1296 append using "${created_data}\NRP LISP.dta"
1297
1298
1299 keep Country Income percentile Score_In_FY21
1300 gen Indicator = "Natural_Resource_Protection"
1301 rename percentile Percentile
1302 save "${created_data}\Natural_Resource_Protection.dta", replace
1303 export excel using "${final_data}\FY21_Natural_Resource_Protection.xlsx", firstrow(variables
) replace
1304
1305 ////////////////////////////////////////////////////
1306 // Child Health //
1307 ////////////////////////////////////////////////////
1308 //Download Instructions:
1309 //This should be contained in the same download as the NRP data above
1310
1311 import excel "${raw_data}\NRPI_CHI.xlsx", sheet("CHI_2020") firstrow clear
1312 rename CountryName Country
1313 keep Country CHI_v2020_19 CHI_v2020_18 CHI_v2020_17 CHI_v2020_16 CHI_v2020_15 CHI_v2020_14
CHI_v2020_13 CHI_v2020_12 CHI_v2020_11 CHI_v2020_10
1314 WBNaming Country
1315
1316 save "${created_data}\Child FY21.dta", replace
1317
1318 **Merging
1319 *****
1320 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1321
1322 save "${created_data}\Child Full.dta", replace
1323 preserve
1324 keep if Income == "LISP"
1325 PercentRankInc CHI_v2020_19
1326 save "${created_data}\Child LISP.dta", replace
1327 restore
1328 keep if Income == "HISP"
1329 PercentRankInc CHI_v2020_19
1330 save "${created_data}\Child HISP.dta", replace
1331 append using "${created_data}\Child LISP.dta"
1332
1333 keep Country Income percentile Score_In_FY21
1334 gen Indicator = "Child_Health"
1335 rename percentile Percentile
1336 save "${created_data}\Child Health.dta", replace
1337 export excel using "${final_data}\FY21_Child_Health.xlsx", firstrow(variables) replace
1338
1339
1340 ////////////////////////////////////////////////////
1341 // Land Rights and Access (IFAD) //
1342 ////////////////////////////////////////////////////
1343
1344 //Download Instructions
1345 /*
1346 Go to https://webapps.ifad.org/members/gc
1347 click on "All Sessions" on the Right to go here https://webapps.ifad.org/members/gc/sessions
1348 Find the most recent Session, and look for "Programme of work and budget of IFAD and its
Office of Evaluation for YEAR"
1349 If this is not present, check a previous Session
1350 Scroll to the bottom of this document to the section titled "YEAR Rural Sector Performance
Assessment (RSPA) scores"
1351 Find the indicator for Access to Land (3.3 in 2018)
1352 Copy all the numbers into a txt file, replace all spaces with commas. Save as a .csv file
1353 Open CSV in excel, the numbers should be in different columns.
1354 Copy the data, use special paste to reformat it so that it is in a single column.

```

```

1355 Copy the country labels in one group at a time. Make sure you use the same order as the
original data.
1356 (You may need to open the file in Adobe Acrobat instead of a web browser to do this).
1357 Delete any cells where the country title is on more than one line. Shift Cells up.
1358 Check to make sure that all of the country data is lined up correctly.
1359 Label the columns "Country" and "Data", Label the sheet IFAD Data
1360
1361 Save as a Excel file (not a CSV) named IFAD Data.xlsx
1362
1363
1364 If data is not available in the current year for a country, MCC pulls forward data from the
previous report (back to 2014 for FY21).
1365 This report can be found by going to https://webapps.ifad.org/members/gc/sessions,
selecting the next previous session (check for the 2015 RSPA at the bottom of the document
1366 then selecting the agenda item with the "IFAD's 2017 results-based programme of work and
regular and capital budgets, the IOE results-based work programme and budget for 2017 and
indicative plan for 2018-2019, and the HIPC and PBAS progress reports"
1367 Once again scroll to the bottom of the document and repeat the steps above for the
indicator Access to Land (B.i in this document). This should be the 2015 Rural Sector
Performance Assessment
1368 Label the columns "Country" and "Data", label the sheet IFAD Data
1369 Save as IFAD_2015.xlsx
1370
1371 Repeat this process, for the 2015 session (38th) of the governing council. These numbers
should be from the 2014 RSPA.
1372 Save as IFAD_2014.xlsx
1373
1374 */
1375
1376 set more off
1377 import excel "${raw_data}\IFAD Data.xlsx", clear firstrow sheet("IFAD Data")
1378 WBNaming Country
1379 rename Data Y2020
1380 save "${created_data}\IFAD_Data.dta", replace
1381
1382 import excel "${raw_data}\IFAD_2015.xlsx", clear firstrow sheet("IFAD Data")
1383 WBNaming Country
1384 rename Data Y2015
1385 merge 1:1 Country using "${created_data}\IFAD_Data.dta", nogen
1386 replace Y2020 = Y2015 if missing(Y2020)
1387 save "${created_data}\IFAD_Data.dta", replace
1388
1389 import excel "${raw_data}\IFAD_2014.xlsx", clear firstrow sheet("IFAD Data")
1390 WBNaming Country
1391 rename Data Y2014
1392 merge 1:1 Country using "${created_data}\IFAD_Data.dta", nogen
1393 replace Y2020 = Y2014 if missing(Y2020)
1394
1395
1396 egen Max_Score2020 = max(Y2020)
1397 egen Min_Score2020 = min(Y2020)
1398 gen IFAD_Score2020 = 1-((Max_Score2020-Y2020)/(Max_Score2020-Min_Score2020))
1399
1400 //keep Country IFAD_Score*
1401 save "${created_data}\IFAD_Score_Normalized.dta", replace
1402
1403
1404 //This indicator is only half of the Land Rights and Access Indicator
1405 //We need to add in the Doing Business Registering Property data in order to create the
full indicator.
1406
1407
1408
1409 ///////////////////////////////////////////////////
1410 // Doing Business //
1411 ///////////////////////////////////////////////////
1412
1413 //Download Instructions:
1414 /*
1415 Note that MCC used old Doing Business Data in FY21 due to concerns explained here:

```



```

1416 https://www.mcc.gov/who-we-select/indicators/doing-business-indicators-fy21
1417
1418 Go to https://www.doingbusiness.org/en/data and select historical data create a custom
dataset
1419 This should take you to: https://www.doingbusiness.org/en/custom-query
1420 Choose all Economies
1421 Choose Topics "Starting a Business" "Getting Credit" and "Registering Property"
1422 Select years back to 2015
1423 Click Create report
1424 Click the excel button to download a report
1425 Save as the file below
1426 */
1427
1428
1429 use "${raw_data}\Historical-data.dta" , clear
1430 keep startbustime startbuscost startbustimeW startbuscostW registertime registercost
creditinformation8pnt5pr creditrights_new economy dbyear

1431
1432 rename startbustime Time_Men_days
1433 rename startbuscost Cost_Men_pc_of_income_pera
1434 rename startbustimeW Time_Women_days
1435 rename startbuscostW Cost_Women_pc_of_income_pi
1436 rename registertime Time_days
1437 rename registercost Cost_pc_of_property_value
1438 rename creditrights_new Strength_of_legal_rights_1
1439 rename creditinformation8pnt5pr Depth_of_credit_informatiD
1440 rename dbyear Year
1441 rename economy Economy
1442 //Replacing Year to Match the actual year of data collection
1443 replace Year = Year - 1
1444
1445 drop in 1
1446
1447 //Keeping only the necessary variables and observations
1448 drop if Year < 2014
1449 drop if missing(Year) & Economy == ""
1450
1451 //Setting variable names up so that they can be reshaped, and reshaping the data to a wide
format
1452 unab Allvars : _all
1453 unab exclude : Economy Year
1454 local Addlist1 : list Allvars - exclude
1455
1456 foreach var of varlist `Addlist1' {
1457 rename `var' `var'_
1458 local `var'lb : variable label `var'
1459 }
1460
1461 unab Allvars : _all
1462 unab exclude : Economy Year
1463 local Addlist2 : list Allvars - exclude
1464 reshape wide `Addlist2' , i(Economy) j(Year)
1465
1466 //Labeling Variables
1467 foreach var of local Addlist1 {
1468 forvalues k = 2014(1)2019 {
1469 label var `var'_`k' "`var'lb' `k'"
1470 }
1471 }
1472
1473 //Generating a maximum for each variable, and putting it in place of the "No Practice" values
1474 unab Allvars : _all
1475 unab exclude : Economy
1476 local Addlist3 : list Allvars - exclude
1477
1478 foreach var of varlist `Addlist3' {
1479 egen x`var' = max(`var')
1480 replace `var' = x`var' if `var' == -9999
1481 drop x`var'
1482 }

```

```

1483
1484 //Merging
1485 WBNaming Economy
1486 rename Economy Country
1487 label var Country "Country"
1488 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1489 drop GNI
1490 save "${created_data}\All_Doing_Business_Formatted.dta" , replace
1491
1492 //This file will be used in what follows to create three indicators: Business Start Up,
1493 Access to Credit, and (with IFAD) Land Rights and Access.
1494
1495 ////////////////////////////////////////////////////
1496 // Access to Credit //
1497 ////////////////////////////////////////////////////
1498 use "${created_data}\All_Doing_Business_Formatted.dta", clear
1499 keep Country Strength_of_legal_rights_1_20* Depth_of_credit_informatiD_20*
1500 //The following computes access to credit scores for a given country by normalizing both
1501 components and then taking an average.
1502 forvalues k=14(1)19 {
1503   gen Norm_Depth_`k' = Depth_of_credit_informatiD_20`k' * 12
1504   gen Norm_Legal_`k' = Strength_of_legal_rights_1_20`k' * 8
1505   gen Access_score_`k' = (Norm_Depth_`k' + Norm_Legal_`k')/2
1506 }
1507 save "${created_data}\Access_Calculated.dta" , replace
1508
1509 **Merging
1510 *****
1511
1512 use "${created_data}\Access_Calculated.dta", clear
1513 keep Country Access_score_19
1514 rename Access_score_19 Access_score
1515 WBNaming Country
1516 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1517
1518 save "${created_data}\Access_Full.dta", replace
1519 preserve
1520 keep if Income == "LISP"
1521 PercentRankInc Access_score
1522 save "${created_data}\Access_LISP.dta", replace
1523 restore
1524 keep if Income == "HISP"
1525 PercentRankInc Access_score
1526 save "${created_data}\Access_HISP.dta", replace
1527 append using "${created_data}\Access_LISP.dta"
1528
1529 keep Country Income percentile Score_In_FY21
1530 gen Indicator = "Access_To_Credit"
1531 rename percentile Percentile
1532 save "${created_data}\Access_To_Credit.dta", replace
1533 export excel using "${final_data}\FY21_Access_To_Credit.xlsx", firstrow(variables) replace
1534
1535
1536 ////////////////////////////////////////////////////
1537 // Business Start-Up //
1538 ////////////////////////////////////////////////////
1539
1540 use "${created_data}\All_Doing_Business_Formatted.dta", clear
1541 keep Country Cost_Men_pc_of_income_pera_20* Cost_Women_pc_of_income_pi_20* Country
1542 Time_Men_days_20* Time_Women_days_20*
1543 forvalues k= 14(1)19 {
1544   egen Time_Average`k' = rowmean(Time_Men_days_20`k' Time_Women_days_20`k')
1545   egen Cost_Average`k' = rowmean(Cost_Men_pc_of_income_pera_20`k'
1546   Cost_Women_pc_of_income_pi_20`k')
1547 //The following finds the min and max of these variables
1548   egen Max_Time`k' = max(Time_Average`k')
1549   egen Max_Cost`k' = max(Cost_Average`k')
1550   egen Min_Time`k' = min(Time_Average`k')

```

```

1549 egen Min_Cost`k' = min(Cost_Average`k')
1550 //The following normalizes these variables to a common scale using the min and max
1551 gen Time_Score`k' = (Max_Time`k'-Time_Average`k')/(Max_Time`k'-Min_Time`k')
1552 gen Cost_Score`k' = (Max_Cost`k'-Cost_Average`k')/(Max_Cost`k'-Min_Cost`k')
1553 gen Business_score`k' = (Time_Score`k' + Cost_Score`k')/2
1554 }
1555 keep Country Time_Score* Cost_Score* Business_score*
1556 save "${created_data}\Business Calculated.dta" , replace
1557
1558 **Merging
1559 *****
1560 use "${created_data}\Business Calculated.dta", clear
1561 keep Country Business_score_19
1562 rename Business_score_19 Business_score
1563 WBNaming Country
1564 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1565
1566 save "${created_data}\Business Full.dta", replace
1567 preserve
1568 keep if Income == "LISP"
1569 PercentRankInc Business_score
1570 save "${created_data}\Business LISP.dta", replace
1571 restore
1572 keep if Income == "HISP"
1573 PercentRankInc Business_score
1574 save "${created_data}\Business HISP.dta", replace
1575 append using "${created_data}\Business LISP.dta"
1576
1577 keep Country Income percentile Score_In_FY21
1578 gen Indicator = "Business_Start_Up"
1579 rename percentile Percentile
1580 save "${created_data}\Business Start Up.dta", replace
1581 export excel using "${final_data}\FY21_Business_Start_Up.xlsx", firstrow(variables) replace
1582
1583
1584 ////////////////////////////////////////////////////
1585 // Land Rights and Access (WB) //
1586 ////////////////////////////////////////////////////
1587
1588 set more off
1589 use "${created_data}\All_Doing_Business_Formatted.dta", clear
1590 keep Country Time_days_20* Cost_pc_of_property_value_20*
1591
1592 forvalues k= 14(1)19 {
1593 egen Max_Time`k' = max(Time_days_20`k')
1594 egen Max_Cost`k' = max(Cost_pc_of_property_value_20`k')
1595 egen Min_Time`k' = min(Time_days_20`k')
1596 egen Min_Cost`k' = min(Cost_pc_of_property_value_20`k')
1597 gen Time_Score`k' = (Max_Time`k'-Time_days_20`k')/(Max_Time`k'-Min_Time`k')
1598 gen Cost_Score`k' = (Max_Cost`k'-Cost_pc_of_property_value_20`k')/(Max_Cost`k'-Min_Cost`k')
1599 gen Rg_Prpr_score`k' = (Time_Score`k' + Cost_Score`k')/2
1600 }
1601
1602 **Merging
1603 *****
1604 merge 1:1 Country using "${created_data}\IFAD_Score_Normalized.dta"
1605 gen LRA_score = (Rg_Prpr_score_19 + IFAD_Score2020)/2
1606
1607 WBNaming Country
1608 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1609
1610 save "${created_data}\LRA Full.dta", replace
1611 preserve
1612 keep if Income == "LISP"
1613 PercentRankInc LRA_score
1614 save "${created_data}\LRA LISP.dta", replace
1615 restore
1616 keep if Income == "HISP"
1617 PercentRankInc LRA_score
1618 save "${created_data}\LRA HISP.dta", replace

```

```

1619 append using "${created_data}\LRA LISP.dta"
1620
1621 keep Country Income percentile Score_In_FY21
1622 gen Indicator = "Land_Rights_And_Access"
1623 rename percentile Percentile
1624 save "${created_data}\Land Rights And Access.dta", replace
1625 export excel using "${final_data}\FY21_Land_Rights_And_Access.xlsx", firstrow(variables)
replace

1626
1627
1628 //////////////////////////////////////
1629 // WGI Control of Corruption //
1630 //////////////////////////////////////
1631 //The following four indicators are drawn from the World Governance Indicators.
1632 //The code for each follows the same procedures
1633 /*
1634 //Download instructions:
1635 Go to https://info.worldbank.org/governance/wgi/ and download the excel file
1636 Name the file wgidataset.xlsx and save in the raw data folder
1637 */
1638
1639 import excel "${raw_data}\wgidataset.xlsx", sheet("ControlofCorruption") clear
1640 forvalues k = 1(1)13 { //Dropping extraneous header information
1641 drop in 1
1642 }
1643 foreach var of varlist _all {
1644 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1645 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1646 replace `var' = "Country" if `var' == "Country/Territory"
1647 }
1648 drop in 2 //Dropping type now that it is incorporated into the first row
1649 renvars, map (word(@[1],1))
1650 drop in 1
1651 destring, replace
1652 save "${created_data}\Control of Corruption Full.dta" , replace
1653
1654 **Merging
1655 *****
1656
1657 keep Country Estimate2019
1658 rename Estimate2019 Corruption_Score
1659
1660 WBNaming Country
1661 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1662 save "${created_data}\Corruption Full.dta", replace
1663
1664 //Calculating Percentile Ranks
1665 preserve
1666 keep if Income == "LISP"
1667 //For all the WGI indicators, we shift all values so that 0 is the median
1668 egen median = median(Corruption_Score)
1669 replace Corruption_Score = Corruption_Score - median
1670 drop median
1671 PercentRankInc Corruption_Score
1672 save "${created_data}\Corruption LISP.dta", replace
1673 restore
1674 keep if Income == "HISP"
1675 //For all the WGI indicators, we shift all values so that 0 is the median
1676 egen median = median(Corruption_Score)
1677 replace Corruption_Score = Corruption_Score - median
1678 drop median
1679 PercentRankInc Corruption_Score
1680 save "${created_data}\Corruption HISP.dta", replace
1681 append using "${created_data}\Corruption LISP.dta"
1682
1683 keep Country Income percentile Score_In_FY21
1684 gen Indicator = "Control_Of_Corruption"
1685 rename percentile Percentile
1686 save "${created_data}\Control Of Corruption.dta", replace

```

```

1687 export excel using "${final_data}\FY21_Control_Of_Corruption.xlsx", firstrow(variables)
      replace
1688
1689
1690 ////////////////////////////////////////////////////
1691 // WGI Government Effectiveness //
1692 ////////////////////////////////////////////////////
1693
1694 import excel "${raw_data}\wgidataset.xlsx", sheet("GovernmentEffectiveness") clear
1695 forvalues k = 1(1)13 { //Dropping extraneous header information
1696 drop in 1
1697 }
1698 foreach var of varlist _all {
1699 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1700 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1701 replace `var' = "Country" if `var' == "Country/Territory"
1702 }
1703 drop in 2 //Dropping type now that it is incorporated into the first row
1704 renvars, map (word(@[1],1))
1705 drop in 1
1706 destring, replace
1707 save "${created_data}\Government Effectiveness Full.dta" , replace
1708
1709 **Merging
1710 *****
1711
1712 keep Country Estimate2019
1713 rename Estimate2019 Gov_Eff_Score
1714
1715 WBNaming Country
1716 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1717
1718 //Calculating Percentiles
1719 preserve
1720 keep if Income == "LISP"
1721 //For all the WGI indicators, we shift all values so that 0 is the median
1722 egen median = median(Gov_Eff_Score)
1723 replace Gov_Eff_Score = Gov_Eff_Score - median
1724 drop median
1725 PercentRankInc Gov_Eff_Score
1726 save "${created_data}\Gov_Eff LISP.dta", replace
1727 restore
1728 keep if Income == "HISP"
1729 //For all the WGI indicators, we shift all values so that 0 is the median
1730 egen median = median(Gov_Eff_Score)
1731 replace Gov_Eff_Score = Gov_Eff_Score - median
1732 drop median
1733 PercentRankInc Gov_Eff_Score
1734 save "${created_data}\Gov_Eff HISP.dta", replace
1735 append using "${created_data}\Gov_Eff LISP.dta"
1736
1737 keep Country Income percentile Score_In_FY21
1738 gen Indicator = "Government Effectiveness"
1739 rename percentile Percentile
1740 save "${created_data}\Government Effectiveness.dta", replace
1741 export excel using "${final_data}\FY21_Government_Effectiveness.xlsx", firstrow(variables)
      replace
1742
1743
1744 ////////////////////////////////////////////////////
1745 // WGI Rule of Law //
1746 ////////////////////////////////////////////////////
1747
1748 import excel "${raw_data}\wgidataset.xlsx", sheet("RuleofLaw") clear
1749 forvalues k = 1(1)13 { //Dropping extraneous header information
1750 drop in 1
1751 }
1752 foreach var of varlist _all {
1753 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,

```

```

Standard Deviation, etc.) into a single variable
1754 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1755 replace `var' = "Country" if `var' == "Country/Territory"
1756 }
1757 drop in 2 //Dropping type now that it is incorporated into the first row
1758 renvars, map (word(@[1],1))
1759 drop in 1
1760 destring, replace
1761 save "${created_data}\Rule of Law Full.dta" , replace
1762
1763 keep Country Estimate2019
1764 rename Estimate2019 RuleofLaw_Score
1765
1766 WBNaming Country
1767 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1768
1769 //Calculate percentiles
1770 preserve
1771 keep if Income == "LISP"
1772 //For all the WGI indicators, we shift all values so that 0 is the median
1773 egen median = median(RuleofLaw_Score)
1774 replace RuleofLaw_Score = RuleofLaw_Score - median
1775 drop median
1776 PercentRankInc RuleofLaw_Score
1777 save "${created_data}\RuleofLaw LISP.dta", replace
1778 restore
1779 keep if Income == "HISP"
1780 //For all the WGI indicators, we shift all values so that 0 is the median
1781 egen median = median(RuleofLaw_Score)
1782 replace RuleofLaw_Score = RuleofLaw_Score - median
1783 drop median
1784 PercentRankInc RuleofLaw_Score
1785 save "${created_data}\RuleofLaw HISP.dta", replace
1786 append using "${created_data}\RuleofLaw LISP.dta"
1787
1788 keep Country Income percentile Score_In_FY21
1789 gen Indicator = "Rule_Of_Law"
1790 rename percentile Percentile
1791 save "${created_data}\Rule Of Law.dta", replace
1792 export excel using "${final_data}\FY21_Rule_Of_Law.xlsx", firstrow(variables) replace
1793
1794 ////////////////////////////////////////////////////
1795 // WGI Regulatory Quality //
1796 ////////////////////////////////////////////////////
1797
1798 import excel "${raw_data}\wgidataset.xlsx", sheet("RegulatoryQuality") clear
1799 forvalues k = 1(1)13 { //Dropping extraneous header information
1800 drop in 1
1801 }
1802 foreach var of varlist _all {
1803 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1804 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1805 replace `var' = "Country" if `var' == "Country/Territory"
1806 }
1807 drop in 2 //Dropping type now that it is incorporated into the first row
1808 renvars, map (word(@[1],1))
1809 drop in 1
1810 destring, replace
1811 save "${created_data}\Regulatory Quality Full.dta" , replace
1812
1813 keep Country Estimate2019
1814 rename Estimate2019 RegQuality_Score
1815
1816 WBNaming Country
1817 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
1818
1819 //Calculating Percentiles
1820 save "${created_data}\RegQuality Full.dta", replace
1821 preserve

```

```

1822 keep if Income == "LISP"
1823 //For all the WGI indicators, we shift all values so that 0 is the median
1824 egen median = median(RegQuality_Score)
1825 replace RegQuality_Score = RegQuality_Score - median
1826 drop median
1827 PercentRankInc RegQuality_Score
1828 save "${created_data}\RegQuality LISP.dta", replace
1829 restore
1830 keep if Income == "HISP"
1831 //For all the WGI indicators, we shift all values so that 0 is the median
1832 egen median = median(RegQuality_Score)
1833 replace RegQuality_Score = RegQuality_Score - median
1834 drop median
1835 PercentRankInc RegQuality_Score
1836 drop median
1837 save "${created_data}\RegQuality HISP.dta", replace
1838 append using "${created_data}\RegQuality LISP.dta"
1839
1840 keep Country Income percentile Score_In_FY21
1841 gen Indicator = "Regulatory_Quality"
1842 rename percentile Percentile
1843 save "${created_data}\Regulatory_Quality.dta", replace
1844 export excel using "${final_data}\FY21_Regulatory_Quality.xlsx", firstrow(variables) replace
1845
1846
1847
1848
1849 ////////////////////////////////////////////////////////////////////
1850 // Freedom of Information //
1851 ////////////////////////////////////////////////////////////////////
1852 // The Freedom of the Press indicator is a composite of three sub indicators
1853 // The main sub indicator is Freedom of the Press, with two minor sub indicators focused on
1854 //Right to Information, and Internet shutdowns
1855 //The following code imports, cleans and combines these files to create scores for each
1856 //country.
1857
1858 ////////////////////////////////////////////////////////////////////
1859 // Freedom of the Press //
1860 ////////////////////////////////////////////////////////////////////
1861 // In previous years, the Freedom of the Press Indicator was drawn from Freedom House's
1862 //Freedom of the Press Report
1863 // However, Freedom House stopped producing this indicator in 2017. This required MCC to
1864 //find another source for these data
1865 // This year, MCC began using the Reporters Without Borders (RSF) Press Freedom Index in
1866 //place of Freedom House.
1867 // However, RSF was missing data for several small island nations, therefore MCC uses
1868 //percentile matching as shown below
1869 // to impute RSF scores for these countries based on data from previous years of Freedom
1870 //House.
1871 //Download Instructions:
1872 /*
1873 Go to the RSP website. Go to the Index details (https://rsf.org/en/ranking\_table), and
1874 select download the dataset
1875 Save as "RSF dataset 2019.csv"
1876 For the historical files: the "index_format_upload 2017.csv" can be found at the bottom of
1877 the page here "https://rsf.org/en/2017-press-freedom-index-ever-darker-world-map"
1878 And the classment2016 file can be found at the bottom of this page:
1879 "https://rsf.org/en/2016-world-press-freedom-index-leaders-paranoid-about-journalists"
1880 Go to the Freedom House website. Go to the Freedom of the Press Report
1881 (https://freedomhouse.org/report-types/freedom-press), and select "Freedom of the Press Data"
1882 Save as FOTP1980-FOTP2017_Public-Data.xlsx
1883 */
1884
1885 // The Following Sections format the RSF files for the Press Freedom Index
1886 //RSF Data Formatting
1887 set more off
1888 //RSF 2020
1889 import delimited "${raw_data}\RSF Data\RSF Dataset 2020.csv", clear
1890 keep en_country score2020
1891 rename en_country Country

```

```

1881 rename score2020 RSFscore
1882
1883 replace RSFscore = substr(RSFscore,"",".",.)
1884 destring, replace
1885 WBNaming Country
1886 save "${created_data}\RSF_Scores_2020.dta", replace
1887
1888 //RSF 2019
1889 import delimited "${raw_data}\RSF Data\RSF Dataset 2019.csv", clear
1890 keep en_country score2019
1891 rename en_country Country
1892 rename score2019 RSFscore
1893
1894 replace RSFscore = substr(RSFscore,"",".",.)
1895 destring, replace
1896 WBNaming Country
1897 save "${created_data}\RSF_Scores_2019.dta", replace
1898
1899 //RSF 2018
1900 import delimited "${raw_data}\RSF Data\RSF Dataset 2019.csv", clear
1901 keep en_country score2018
1902 rename en_country Country
1903 rename score2018 RSFscore
1904
1905 replace RSFscore = substr(RSFscore,"",".",.)
1906 destring, replace
1907 WBNaming Country
1908 save "${created_data}\RSF_Scores_2018.dta", replace
1909
1910 //RSF 2017
1911 import delimited "${raw_data}\RSF Data\index_format_upload 2017.csv", delimiter(";") clear
1912 //This file is mislabeled. The data that is labeled as "overallscore2016" is actually 2017
data (according to the main map). The score2015 data is actually 2016 data (as it matches
data from the 2016 file).
1913 keep en_country overallscore2016 //This is actually 2017 data
1914 rename en_country Country
1915 rename overallscore2016 RSFscore
1916
1917 replace RSFscore = substr(RSFscore,"",".",.)
1918 destring, replace
1919 WBNaming Country
1920 save "${created_data}\RSF_Scores_2017.dta", replace
1921
1922 //RSF 2016
1923 import delimited "${raw_data}\RSF Data\index_format_upload 2017.csv", delimiter(";") clear
1924 //This file is mislabeled. The data that is labeled as "overallscore2016" is actually 2017
data (according to the main map). The score2015 data is actually 2016 data (as it matches
data from the 2016 file).
1925 keep en_country score2015 //This is actually 2016 data
1926 rename en_country Country
1927 rename score2015 RSFscore
1928
1929 replace RSFscore = substr(RSFscore,"",".",.)
1930 destring, replace
1931 WBNaming Country
1932 save "${created_data}\RSF_Scores_2016.dta", replace
1933
1934 //RSF 2015
1935 import delimited "${raw_data}\RSF Data\classement2016.csv", delimiter(";") clear
1936 keep en_country score2015
1937 rename en_country Country
1938 rename score2015 RSFscore
1939
1940 replace RSFscore = substr(RSFscore,"",".",.)
1941 destring, replace
1942 WBNaming Country
1943 save "${created_data}\RSF_Scores_2015.dta", replace
1944
1945
1946 //The following section formats the Freedom House data

```



```

1947 //FH Data Formatting
1948 import excel "${raw_data}\FOTP1980-FOTP2017_Public-Data.xlsx", clear sheet(Data)
1949 replace A = "Country" in 5
1950 //The following section fills in missing year values for observations which were merged in
the original document
1951 qui ds //List all variables
1952 local variable_tally : word count `r(varlist)' //Count the number of variables
1953 local tally_minus = `variable_tally' - 1 //Create a macro which is one less than the full
variable count
1954 forvalues k = 1(1)`tally_minus' {
1955 local left : word `k' of `r(varlist)' //Finding the "left" value for the year observation
1956 local right : word `='k'+1' of `r(varlist)' //Finding the "right" value for the year
observation
1957 replace `right' = `left' if `right'[3] == "" in 3 //Filling forward the year values when
the values are missing
1958 }
1959 //Dropping Statuses and Sub-Scores
1960 foreach var of varlist _all {
1961 if !(`var'[5] == "Print" | `var'[5] == "Broadcast" | `var'[5] == "Status" | `var'[5] ==
"Total Score" | `var'[5] == "Country") {
1962 drop `var'
1963 }
1964 }
1965
1966 //Dropping extraneous headings
1967 drop in 4
1968 drop in 2
1969 drop in 1
1970 replace A = "Country" in 1
1971 //The following section of code combines the year headings with the data type headings
1972 foreach var of varlist _all {
1973 label var `var' "`='var'[1]'"
1974 replace `var' = "Y`='var'[1]`_'_`='var'[2]'" in 1
1975 replace `var' = substr("`='var'[1]'", "-", "_", .) in 1
1976 }
1977 replace A = "Country" in 1
1978
1979 renvars, map (word(@[1],1))
1980 drop in 1
1981 drop in 1
1982
1983 destring, replace
1984 WBNaming Country
1985 save "${created_data}\Freedom_House_Press_Formatted.dta", replace
1986
1987
1988 **Imputation
1989 *****
1990 //Now that we have formatted the two files, we need to merge them and
1991 //use the freedom house data to impute the new scores using percentile
1992 //matching
1993
1994 set more off
1995 //This loop takes each RSF year and uses the freedom house data to impute the correct scores.
1996 forvalues RSF_year = 2015(1)2020 {
1997
1998 use "${created_data}\Freedom_House_Press_Formatted.dta", clear
1999
2000 merge 1:1 Country using "${created_data}\RSF_Scores_`RSF_year'.dta", nogen keep(1 3)
2001 //Since the most recent Freedom House data is from 2017, when the RSF year is 2017, 2018,
or 2019,
2002 // we use that Freedom House data to impute the scores. However in earlier years,
2003 // we use matching year data:
2004 if "`RSF_year'" == "2020" | "`RSF_year'" == "2019" | "`RSF_year'" == "2018" | "`RSF_year'" ==
"2017" {
2005 rename Y2017_Total FHScore
2006
2007 }
2008 else {
2009 rename Y`RSF_year'_Total FHScore

```

```

2010 }
2011
2012 replace FHScore = "." if FHScore == "--"
2013 keep Country RSFscore FHScore
2014 destring, replace
2015 //In this line we are making the decision to only consider countries that are in the GNI
dataset as described above.
2016 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
2017 //In order to calculate the percentile matching scores, we need to calculate the number
2018 //of Freedom House Scores in the full dataset.
2019 egen n = count(FHScore)
2020 //We also need to calculate a ranking (1+ the number of scores that are lower)
2021 egen i = rank(FHScore), track
2022 //The Percentile rank in the full Freedom House dataset is then calculated as the rank
minus 1, divided by the count minus 1
2023 gen pcrank_FH = (i - 1) / (n - 1)
2024 drop i n
2025 egen RSF_count = count(RSFscore)
2026 //Now, we use the following calculations to determine where the rank of the freedom house
percentile
//in the RSF dataset, and split it into its whole number and decimal part
2027 gen FH_Rank_in_RSF = (pcrank_FH*(RSF_count-1))+1
2028 gen FH_floor = floor(FH_Rank_in_RSF)
2029 gen FH_decimal = FH_Rank_in_RSF - FH_floor
2030
2031
2032 sort RSFscore
2033 local RSF_cnt = `=RSF_count[1]`
2034 //This code creates a set of global macros corresponding to the ranking of the RSF scores
2035 qui forvalues k=1(1)`RSF_cnt' {
2036 local k_plus = `k' + 1
2037 global RSF_function_`k' = `=RSFscore[`k]`
2038 noisily di "${RSF_function_`k'} `k'"
2039 }
2040 gen function_x = .
2041 gen function_x_plus = .
2042 local count = _N
2043 //The following code finds the RSF scores which rank on either side of
2044 //The Freedom House ranking, and then adding the product of the decimal
2045 //part of the Freedom House ranking and the difference of the two RSF scores
2046 //on either side.
2047 qui forvalues k = 1(1)`count' {
2048 replace function_x = ${RSF_function_`=FH_floor[`k]`} in `k'
2049 local floor_plus = `=FH_floor[`k]` + 1
2050 local count_minus = `count' -1
2051 if "`=FH_floor[`k]`" == "`RSF_cnt'" {
2052 noi di "`k'"
2053 local floor_plus = `=FH_floor[`k]`
2054 noisily di "${RSF_function_`=floor_plus}"
2055 }
2056 replace function_x_plus = ${RSF_function_`=floor_plus`} in `k'
2057 }
2058 gen New_RSF_Score = function_x + (FH_decimal*(function_x_plus-function_x))
2059 //This replaces the missing RSF scores with the newly imputed scores based on the FH data
2060 replace RSFscore = New_RSF_Score if missing(RSFscore)
2061 drop New_RSF_Score function_x_plus function_x FH_decimal FH_floor FH_Rank_in_RSF RSF_count
pcrank_FH GNI FHScore
2062 rename RSFscore Y`RSF_year'_Total
2063 label var Y`RSF_year'_Total "`RSF_year'"
2064 gen Y`RSF_year'_Status = "NF"
2065 label var Y`RSF_year'_Status "`RSF_year'"
2066 WBNaming Country
2067 save "${created_data}\Freedom_of_The_Press_Imputed_`=RSF_year'.dta", replace
2068 }
2069 //Merging each of the year files together
2070 use "${created_data}\Freedom_of_The_Press_Imputed_2015.dta", clear
2071 forvalues RSF_year = 2016(1)2019 {
2072 merge 1:1 Country using "${created_data}\Freedom_of_The_Press_Imputed_`=RSF_year'.dta", nogen
2073 }
2074 save "${created_data}\FOP FY21.dta", replace
2075

```

```

2076 //Before we can create the final file, we need to process the Key Internet Controls and the
      Right to Information files.
2077
2078 ////////////////////////////////////////////////////
2079 //      Access Now      //
2080 ////////////////////////////////////////////////////
2081 //Download instructions:
2082 //Download from https://www.accessnow.org/keepiton/ by clicking "Download the STOP Data"
2083
2084 import delimited
      "${raw_data}\KeepItON-Internet-Shutdown-Data-STOP-Shutdown-Tracker-2019-Complete.csv",
      bindquote(strict) clear
2085 WBNaming country
2086 replace duration = "248" if duration == "248 at the time of publication "
2087 replace duration = "1" if duration == "0" | duration == "2 hours " | /*
2088 */ duration == "For about two hours " | duration == "for a few hours " // | duration == ""
2089
2090 rename end_date DateEnded
2091
2092 replace DateEnded = substr(DateEnded, " ", "", .)
2093 replace DateEnded = start_date if DateEnded == "Unkown" | DateEnded == "Unko"
2094 replace DateEnded = "12/31/2019" if DateEnded == "ongoing" | DateEnded == "Ongoing"
2095 gen length = strlen(DateEnded)
2096 gen first = strpos(DateEnded, "/")
2097 gen second = strrpos(DateEnded, "/")
2098 gen month = substr(DateEnded, 1, first-1)
2099 gen day = substr(DateEnded, first+1, (second-first)-1)
2100 gen year = substr(DateEnded, second+1, 4)
2101 destring, replace
2102 gen DateEnd = mdy(month, day, year)
2103 format DateEnd %td
2104 drop length first second month day year
2105
2106 rename start_date DateBegan
2107 replace DateBegan = substr(DateBegan, " ", "", .)
2108
2109 gen length = strlen(DateBegan)
2110 gen first = strpos(DateBegan, "/")
2111 gen second = strrpos(DateBegan, "/")
2112 gen month = substr(DateBegan, 1, first-1)
2113 gen day = substr(DateBegan, first+1, (second-first)-1)
2114 gen year = substr(DateBegan, second+1, 4)
2115 destring, replace
2116 gen DateStart = mdy(month, day, year)
2117 format DateStart %td
2118 gen duration_new = DateEnd - DateStart
2119 replace duration_new = 1 if duration_new == 0 | duration_new == .
2120 compare duration duration_new
2121 replace duration = duration_new if missing(duration)
2122 tab duration
2123
2124 drop if ordered_by == "Non-government"
2125
2126 destring, replace
2127 gen count = 1
2128 collapse (sum) count duration, by (country)
2129
2130 gen AN_Binary = 1
2131 gen AN_Duration = duration
2132 rename country Country
2133 keep Country AN_Binary AN_Duration
2134 gen Internet_Shutdowns = AN_Duration
2135 replace Internet_Shutdowns = 9 if Internet_Shutdowns > 9
2136 label var Internet_Shutdowns "Internet Shutdowns (Duration)"
2137 keep Country Internet_Shutdowns
2138 save "${created_data}\Freedom of Internet.dta", replace
2139
2140 //Before we can create the indicator, we need to process the Right to Information file.
2141
2142 ////////////////////////////////////////////////////

```

```

2143 // Right to Information //
2144 ///////////////////////////////////////////////////
2145 //Download Instructions: Go to the Country data page:
https://www.rti-rating.org/country-data/
2146 // Click on "Excel" download. Save as RTI Ratings in the raw data folder
2147 import excel "${raw_data}\RTI Ratings.xlsx", sheet("Sheet1") firstrow clear
2148 keep Country Total
2149 WBNaming Country
2150 save "${created_data}\Right_To_Information.dta", replace
2151
2152
2153 **Freedom of Information Merging
2154 *****
2155 //Merging files together:
2156 use "${created_data}\Right_To_Information.dta", clear
2157 merge 1:1 Country using "${created_data}\Freedom of Internet.dta", nogen
2158 merge 1:1 Country using "${created_data}\Freedom_of_The_Press_Imputed_2020.dta", nogen
2159 keep Country Total Internet_Shutdowns Y2020_Total
2160 gen RTI_score = 4 if !missing(Total)
2161 replace Internet_Shutdowns = 9 if !missing(Internet_Shutdowns) & Internet_Shutdowns>9
2162 recode RTI_score Internet_Shutdowns (.=0)
2163 gen FOI_score = (Y2020_Total - RTI_score) + Internet_Shutdowns
2164
2165 WBNaming Country
2166 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
2167
2168 save "${created_data}\FOI Testing Full.dta", replace
2169 preserve
2170 keep if Income == "LISP"
2171 PercentRankInc FOI_score
2172 replace percentile = 1-percentile //Note this is different for FOI due to the inverted
scale!
2173 save "${created_data}\FOI Testing LISP.dta", replace
2174 export excel using "${testing_data}\FOI Testing LISP.xls", firstrow(variables) replace
2175 restore
2176 keep if Income == "HISP"
2177 PercentRankInc FOI_score
2178 replace percentile = 1-percentile //Note this is different for FOI due to the inverted
scale!
2179 save "${created_data}\FOI Testing HISP.dta", replace
2180 export excel using "${testing_data}\FOI Testing HISP.xls", firstrow(variables) replace
2181 append using "${created_data}\FOI Testing LISP.dta"
2182 save "${created_data}\FOI Testing.dta", replace
2183 gen Indicator = "Freedom_Of_Information"
2184 rename percentile Percentile
2185 keep Country Income Percentile Score_In_FY21 Indicator
2186 export excel using "${final_data}\FY21_Freedom_Of_Information.xlsx", firstrow(variables)
replace
2187
2188
2189 ///////////////////////////////////////////////////
2190 // Trade Policy //
2191 ///////////////////////////////////////////////////
2192 /* Download Instructions:
2193 As of the publication of this do file, Heritage's scores for trade freedom are not public
for all MCC candidate countries
2194 However, when they become public they will be available here
https://www.heritage.org/index/download
2195 Select "Download Raw Data" and save the file as index2021_data.xls in the raw data folder
*/
2196
2197
2198 set more off
2199 import excel "${raw_data}\2021 Trade Freedom MCC.xlsx", clear firstrow
2200 keep Country TradeFreedom
2201 replace TradeFreedom = "." if TradeFreedom == "N/A"
2202 destring, replace
2203 rename TradeFreedom Trade_Policy_Score
2204 WBNaming Country
2205
2206 **Merging

```

```

2207 *****
2208 WBNaming Country
2209 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
2210
2211 save "${created_data}\Trade Policy Full.dta", replace
2212 preserve
2213 keep if Income == "LISP"
2214 PercentRankInc Trade_Policy_Score
2215 save "${created_data}\Trade Policy LISP.dta", replace
2216 restore
2217 keep if Income == "HISP"
2218 PercentRankInc Trade_Policy_Score
2219 save "${created_data}\Trade Policy HISP.dta", replace
2220 append using "${created_data}\Trade Policy LISP.dta"
2221
2222 keep Country Income percentile Score_In_FY21
2223 gen Indicator = "Trade_Policy"
2224 rename percentile Percentile
2225 save "${created_data}\Trade Policy.dta", replace
2226 export excel using "${final_data}\FY21_Trade_Policy.xlsx", firstrow(variables) replace
2227
2228
2229 ////////////////////////////////////////////////////
2230 //      WEO/IMF Data      //
2231 ////////////////////////////////////////////////////
2232 /*
2233 Download instructions
2234 Use the October 2019 release
2235 Go to https://www.imf.org/external/pubs/ft/weo/2019/02/weodata/index.aspx
2236 Select by countries
2237 Select all countries
2238 Select Inflation, average consumer prices (Percent change) and General government net
lending/borrowing (Percent of GDP)
2239 Select Start 1980, end 2024
2240
2241 */
2242 set more off
2243 import delimited "${raw_data}\WEOOct2020all.xls", clear varnames(nonames)
2244 drop v57
2245 foreach var of varlist v10-v55 {
2246
2247     replace `var' = "Y`='var'[1]'" in 1
2248 }
2249 foreach var of varlist _all {
2250     replace `var' = substr("`var'[1]'", " ", "_",.) in 1
2251     replace `var' = substr("`var'[1]'", "/", "_",.) in 1
2252     replace `var' = substr("`var'[1]'", "-", "_",.) in 1
2253
2254     replace `var' = "." if `var' == "n/a"
2255     replace `var' = "." if `var' == "--"
2256     replace `var' = substr(`var',"",",",.)
2257 }
2258 renvars, map (word(@[1],1))
2259 drop in 1
2260 destring, replace
2261 drop Y2020 Y2021 Y2022 Y2023 Y2024 Y2025 Estimates_Start_After
2262 foreach var of varlist Y* {
2263     lab var `var' `=substr("`var'", "Y",",",.)'
2264 }
2265 save "${created_data}\WEO_IMF_Data_Full.dta" , replace
2266
2267
2268 ////////////////////////////////////////////////////
2269 //      Fiscal Policy      //
2270 ////////////////////////////////////////////////////
2271 //This pulls directly from an IMF indicator
2272 set more off
2273 use "${created_data}\WEO_IMF_Data_Full.dta", clear
2274 keep if Subject_Descriptor == "General government net lending/borrowing"
2275 keep if Units == "Percent of GDP"

```

```

2276 keep Country Y*
2277 WBNaming Country
2278
2279 **Merging
2280 *****
2281 //Fiscal Policy is a three year rolling average
2282 egen Fiscal_Policy_Score = rowmean(Y2017 Y2018 Y2019)
2283 keep Country Fiscal_Policy_Score
2284
2285 WBNaming Country
2286 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
2287
2288 save "${created_data}\Fiscal Policy Full.dta", replace
2289 preserve
2290 keep if Income == "LISP"
2291 PercentRankInc Fiscal_Policy_Score
2292 save "${created_data}\Fiscal Policy LISP.dta", replace
2293 restore
2294 keep if Income == "HISP"
2295 PercentRankInc Fiscal_Policy_Score
2296 save "${created_data}\Fiscal Policy HISP.dta", replace
2297 append using "${created_data}\Fiscal Policy LISP.dta"
2298
2299 keep Country Income percentile Score_In_FY21
2300 gen Indicator = "Fiscal_Policy"
2301 rename percentile Percentile
2302 save "${created_data}\Fiscal Policy.dta", replace
2303 export excel using "${final_data}\FY21_Fiscal_Policy.xlsx", firstrow(variables) replace
2304
2305 ///////////////////////////////////////////////////
2306 //      Inflation      //
2307 ///////////////////////////////////////////////////
2308 //This pulls directly from an IMF indicator
2309 set more off
2310 use "${created_data}\WEO_IMF_Data_Full.dta", clear
2311 keep if Subject_Descriptor == "Inflation average consumer prices"
2312 keep if Units == "Percent change"
2313 keep Country Y*
2314 WBNaming Country
2315
2316 **Merging
2317 *****
2318 rename Y2019 Inflation_Rate
2319 keep Country Inflation_Rate
2320
2321 WBNaming Country
2322 merge 1:1 Country using "${created_data}\GNI_FY21.dta", nogen keep(3)
2323
2324 save "${created_data}\Inflation Full.dta", replace
2325 preserve
2326 keep if Income == "LISP"
2327 PercentRankInc Inflation_Rate
2328 replace percentile = 1-percentile //Note this is different for Inflation due to the
inverted scale
2329 save "${created_data}\Inflation LISP.dta", replace
2330 restore
2331 keep if Income == "HISP"
2332 PercentRankInc Inflation_Rate
2333 replace percentile = 1-percentile //Note this is different for Inflation due to the
inverted scale
2334 save "${created_data}\Inflation HISP.dta", replace
2335 append using "${created_data}\Inflation LISP.dta"
2336
2337 keep Country Income percentile Score_In_FY21
2338 gen Indicator = "Inflation"
2339 rename percentile Percentile
2340 save "${created_data}\Inflation.dta", replace
2341 export excel using "${final_data}\FY21_Inflation.xlsx", firstrow(variables) replace
2342
2343

```

2344
2345
2346