

```

1
2 ////////////////////////////////////////////////////////////////////
3 //                MCC FY20 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 FY20 Guide to the Indicators
12 (https://www.mcc.gov/resources/doc/guide-to-the-indicators-fy-2020)
13 and Data Notes
14 (https://www.mcc.gov/resources/doc/pub-fy2020-scorecard-data-notes)
15 This file is included to provide complete clarity as to exactly how
16 these indicators are constructed.
17
18 To use this file, you need to download the appropriate files from the
19 indicator institutions as described below, and set the global macros
20 to the correct locations.
21
22 This file will output individual files for each indicator.  You can run
23 sections of the program individually, but the section on GNI must be run
24 for the rest to work.
25
26 The official final data files are contained on MCC's open data site.
27 The purpose of this file is to provide systematic instructions explaining
28 the methodology used to create these files.  Due to methodological changes
29 and historical data revisions by indicator institutions or MCC, this program
30 is not designed to recreate previous year's scorecards.  Rather it is designed
31 to clearly show MCC's methodology in FY20.  Updates will be made annually as needed.
32 */
33
34 *****
35 ** Preliminary Settings **
36 *****
37 ** The following settings set up the necessary environment for the do file to run
38 ** This includes programs which are run repeatedly in the file.
39
40 set more off
41 clear
42 ////////////////////////////////////////////////////////////////////
43 // Global Macros //
44 ////////////////////////////////////////////////////////////////////
45
46 //To use this file you need to set the "globals" to the correct file paths on your computer
47 //raw_data is where you are storing the raw data, (instructions for download are included
48 in each section below)
49 //created_data is where you would like the program to store temporary files
50 //final_data is where you would like the program to output the final files
51 global raw_data "WRITE_RAW_FILE_PATH_HERE" //Your filepath will probably be something like
52 C:\Users\YOURUSERNAME\Documents\RAWDATAFOLDERNAME\
53 global created_data "WRITE_CREATED_FILE_PATH_HERE" //Your filepath will probably be
54 something like C:\Users\YOURUSERNAME\Documents\CREATEDDATAFOLDERNAME\
55 global final_data "WRITE_FINAL_FILE_PATH_HERE" //Your filepath will probably be something
56 like C:\Users\YOURUSERNAME\Documents\FINALDATAFOLDERNAME\
57
58 ////////////////////////////////////////////////////////////////////
59 // Programs //
60 ////////////////////////////////////////////////////////////////////
61 //The following sub-programs are used throughout the main program
62 //to rename countries and calculate percentiles.
63
64 **WBNaming
65 *****
66 //The following program is used to rename countries to a common set of names
67 //So that they can easily be merged together.

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

319
320
321 **PercentRankInc
322 *****
323 //The following program mimics Excel's PercentRankInc function
324 //For indicators with a static cutoff (Civil Liberties, Political Rights,
325 //Immunization Rates for the higher scorecard pool and Inflation) this is done normally.
326 //However, for indicators with a median based cutoff, countries at the median are set to
327 //the 50th percentile, countries at the maximum as set to the 100th percentile
328 //and countries at the minimum are set to the 0th percentile
329
330 capture program drop PercentRankInc
331 program PercentRankInc
332 args vartest
333
334 generate over = 300 //Dummy value for now, this will be the number of countries scoring
335 generate under = 300 //Dummy value for now, this will be the number of countries scoring
336 global count = _N //Number of observations
337 global vartest = "`vartest'"
338
339 forvalues line_change = 1(1)$count { //This loops once for each country.
340     global overcount = -1 //This counts the number of countries over the country in
341     line_change. It starts at -1 as we do not count the country itself as being greater than
342     or equal to itself.
343     forvalues line_test = 1(1)$count { //This loops over all the countries again,
344     comparing them each to the selected country from line_change
345     if `=${vartest}[\`line_change']' <= `=${vartest}[\`line_test']' &
346     `=${vartest}[\`line_test']' != . { //This if statement augments the overcount whenever a
347     country from line_test is greater than or equal to the selected country from line_change
348     global overcount = ${overcount} + 1
349     }
350     replace over = ${overcount} in `line_change' //This replaces the variable over with the
351     overcount macro
352     replace over = . in `line_change' if missing(`=${vartest}[\`line_change']') //This sets
353     over to missing if the original value is missing.
354 }
355 }
356
357 forvalues line_change = 1(1)$count { //This loops once for each country.
358     global undercount = 0 //This counts the number of countries over the country in
359     line change. It starts at 0 since we are not testing less than or equal to, so we will not
360     count the country against itself.
361     forvalues line_test = 1(1)$count { //This loops over all the countries again, comparing
362     them each to the selected country from line_change
363     if `=${vartest}[\`line_change']' > `=${vartest}[\`line_test']' &
364     `=${vartest}[\`line_test']' != . { //This if statement augments the undercount whenever a
365     country from line_test is less than the selected country from line_change
366     global undercount = ${undercount} + 1
367     }
368     replace under = ${undercount} in `line_change' //This replaces the variable under with
369     the undercount macro
370     replace under = . in `line_change' if missing(`=${vartest}[\`line_change']') //This sets
371     over to missing if the original value is missing.
372 }
373 }
374
375 gen percentile = under/(over+under) //This calculates percentiles for all observations
376 using the under and over variables.
377
378 //The if statement below forces countries at the median to the 50th percentile, at the max
379 to the 100th percentile and at the min to the 0th percentile
380 //For all the median based indicators (see above for more information).
381 if "`vartest'" != "Civil_Liberties" & "`vartest'" != "Political_Rights" & "`vartest'" !=
382 "Inflation" & "`vartest'" != "Immunization_Rate_HISP" {
383     egen median = median(`vartest')
384     egen min = min(`vartest')
385     egen max = max(`vartest')
386     replace percentile = 0.5 if median == `vartest'
387     replace percentile = 1 if max == `vartest'

```

```

370  replace percentile = 0 if min == `vartest'
371  }
372  gen Score_In_FY20 = `vartest'
373  end
374
375  *****
376  ** Import/Clean Raw Data **
377  *****
378  ** The following sections take the necessary data files for MCC's Scorecard,
379  ** import them, clean them and save the necessary files. This section
380  ** includes instructions for downloading these files as well as processing them.
381  ** in comments before each section.
382
383  ///////////////
384  // GNI //
385  ///////////////
386  //Countries are divided into two scorecard pools based on GNI per capita.
387  //The following is the construction of that measure from World Bank data.
388
389  //Download Instructions:
390  /*
391  Download the xls file from:
392  https://datacatalog.worldbank.org/dataset/gni-capita-ranking-atlas-method-and-ppp-based
393  Check if columns and sheet names still match
394  Check if the footnotes match the income categories (When there is no GNI data
395  available for a country, in some cases the World Bank still provides a determination
396  of which income group that country would be in, there are captured in footnotes in the
  GNIPC file.)
397
398  Save in the ${raw_data} folder as GNIPC.xls
399
400  Note that MCC uses the July release. Sometimes the World Bank will change data
401  in between the July and the September release. The version from the July release
402  is available on MCC's Open Data Portal
403  */
404
405  import excel "${raw_data}\GNIPC.xls", sheet("GNIPC") clear
406  //According to the World Bank, these countries fit MCC's definition of LISP and HISP
407  //Therefore the footnotes are translated into income categories
408  //D is country name, E is GNI, F is a footnote for countries missing GNI
409  //Sometimes the world bank will change which footnotes correspond to which income
  categories, check this carefully.
410  //LISP stands for Lower Income Scorecard Pool
411  //HISP stands for Higher Income Scorecard Pool
412  //These are divided by the Historic IDA threshold.
413  replace E = "LISP" if F == "k"
414  replace E = "HISP" if F == "l"
415  replace E = "UMIC" if F == "i"
416  replace E = "HIC" if F == "j"
417  keep D E
418  // The following are not listed as "Independent States" by the State Department,
419  // and therefore do not receive scorecards. In the following sections, we limit the
420  // dataset by only keeping countries that match a country in this dataset.
421
422  drop if D == "" | D == "Economy" | D == "World" | D == "East Asia & Pacific" /*
423  */ | D == "Europe & Central Asia" | D == "Latin America & Caribbean" /*
424  */ | D == "Middle East & North Africa" | D == "North America" | D == "South Asia" /*
425  */ | D == "Sub-Saharan Africa" | D == "Low income" | D == "Lower middle income" /*
426  */ | D == "Upper middle income" | D == "High income" | D == "West Bank and Gaza"
427
428  //These are placeholder values over the threshold for countries without data.
429  replace E = "15000" if E == "HIC"
430  replace E = "6000" if E == "UMIC"
431  replace E = "0" if E == "LISP"
432  replace E = "2500" if E == "HISP"
433  rename D Country
434  rename E GNI
435
436  //Now that all of the values for GNI are numeric, we can create a string variable "Income"
  Sorting them by threshold

```



```

437 //These threshold categories can be found in the following file labeled "Threshold"
438 //They can be different every year, and so will need to be changed for future iterations
439 destring, replace
440 generate Income = "LISP"
441 replace Income = "HISP" if GNI > 1925
442 replace Income = "UMIC" if GNI > 3995
443 replace Income = "HIC" if GNI > 12375
444
445 //This program standardizes all of the country names, so that they can be matched with
other data files
446 WBNaming Country
447
448 save "${created_data}\GNI_FY20.dta", replace
449 export excel using "${final_data}\FY20_GNI.xlsx", firstrow(variables) replace
450
451 ////////////////
452 // Thresholds //
453 ////////////////
454 //Countries are divided into two scorecard income pools based on GNI per capita.
455 //The cutoff for these pools is the Historic IDA threshold from the World Bank
456 //The following section demonstrates how these are constructed from the World Bank Data.
457 //This is where the cutoffs for the Income variable in the previous section come from.
458
459 //Download Instructions:
460 /*
461 Download the historical classification by income in XLS format from
462 https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lendi
ng-groups
463 Check if columns and sheet names still match
464 Save in the ${raw_data} folder as OGHIST.xls
465 */
466 import excel "${raw_data}\OGHIST.xls", sheet("Thresholds") clear
467 keep A S-AU
468 keep if A == "Date:" | A == "Bank's fiscal year:" | A == "II. IDA Eligibility /2 /7" | A ==
"Lower middle income" | A == "Upper middle income"
469 sort A
470 replace A = "Threshold" if A == "Bank's fiscal year:"
471 //You may need to install renvars. Enter "help renvars" and install the update.
472 //This command renames the variables with the first row of observations.
473 renvars, map (word(@[1],1))
474 drop in 1
475 foreach var of varlist _all {
476 replace `var' = strreverse(substr(strreverse(`var'),1,4)) in 1 //Takes the year (last 4
digits) only from each value in the first row
477 replace `var' = substr(`var',(strpos(`var',"=")+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)
478 replace `var' = substr(`var',(strpos(`var'["-")+1),.) in 3 //Isolates the cutoff value
for the third row (the cutoff between LMICs and UMICs)
479 replace `var' = substr(`var',(strpos(`var'[4],"-")+1),.) in 4 //Isolates the cutoff value
for the fourth row (the cutoff between UMICs and HICs)
480 replace `var' = subinstr(`var',"",".",.) //Removes commas from all numbers so they can be
converted to numeric values.
481 }
482 destring, replace
483
484 xpose, clear varname //Reshapes data
485 label variable v1 "Year"
486 label variable v2 "LISP"
487 label variable v3 "HISP"
488 label variable v4 "UMIC"
489 label variable varname "Fiscal Year"
490 drop in 1
491 order _varname
492 save "${created_data}\Threshold_FY20.dta", replace
493 export excel using "${final_data}\FY20_Threshold.xlsx", firstrow(varlabels) replace
494
495 ////////////////
496 // Population //
497 ////////////////

```

```

498 //Population is not used to determine scorecard passage.
499 //However, it is included on the scorecard as a descriptive statistic in the upper left
    corner,
500 // so the method for constructing it is listed below
501
502 //Download Instructions:
503 /*
504 Go to https://data.worldbank.org/indicator/SP.POP.TOTL
505 Download the excel file
506 Save it as World Bank Population Data.xls in the raw data folder
507 */
508 import excel "${raw_data}\World Bank Population Data.xls", sheet("Data") clear
509 //Dropping unnecessary observations with datasource information
510 drop in 3
511 drop in 2
512 drop in 1
513 //Formatting the first row of variables to use renvars
514 foreach var of varlist all{
515 label var `var' "`='var'[1]'"
516 replace `var' = "Y" + `var'[1] in 1
517 }
518 foreach var of varlist _all{
519 replace `var' = substr("`='var'[1]'", " ", "_",.) in 1
520 }
521 renvars, map (word(@[1],1))
522 drop in 1
523 destring, replace
524 rename YCountry_Name Country
525 drop YCountry_Code YIndicator_Name YIndicator_Code
526 WBNaming Country
527 //This takes out all of the things that are not considered "Independent States" by the
    State Department
528 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keepusing() keep(3) //This
    removes those observations which were not included in the GNI data above.
529 drop GNI
530 keep Country Y2019
531 export excel using "${final_data}\FY20 Population.xlsx", firstrow(varlabels) replace
532
533
534 ////////////////////////////////////////////////////////////////////
535 // Democratic Rights Indicators (FH) //
536 ////////////////////////////////////////////////////////////////////
537 //The Democratic Rights Indicators are Political Rights and Civil Liberties
538 //They are both sourced from Freedom House's Freedom In The World report.
539 //Download Instructions
540 /*
541 Download the Aggregate Category and Subcategory Scores, 2003-2019 (Excel) file from
542 https://freedomhouse.org/report-types/freedom-world
543 Check if columns and sheet names still match the lines below (i.e. is the sheet still
544 named "FIW06-19" and the variable for country/territory still named CT)
545 Save in the ${raw_data} folder as Aggregate Category and Subcategory Scores FIW2003-2018.xlsx
546 */
547 set more off
548 import excel "${raw_data}\Aggregate_Category_and_Subcategory_Scores_FIW2019.xls", sheet(
    "FIW06-19") firstrow clear
549 keep if CT == "c" //Only keeping countries
550 keep if Edition == 2019 //Only keeping current year
551 keep CountryTerritory PR CL //Only keeping country name and relevant data values
    (PR=Political Rights, CL=Civil Liberties)
552 rename CountryTerritory Country
553
554 **To Merge with GNI data
555 *****
556 WBNaming Country //Renaming countries to match GNI data
557 save "${created_data}\FH_FY20.dta", replace
558
559 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3) //This only keeps
    countries included in the GNI data
560
561 keep Country Income CL PR

```

```

562 rename CL Civil_Liberties
563 rename PR Political_Rights
564 save "${created_data}\FH Full.dta", replace
565
566 //Percentile are not used for determining passage on this indicator, but they are included
on the scorecard
567 //The following computes the percentile ranks for the Democratic Rights Indicators
568 //It is worthy of note that MCC's general rule of forcing countries at the median to the
50th percentile and
569 // countries at the max to the 100th percentile does not apply here since percentiles are
not used in the calculation of passage
570
571 global FH_vars "Civil_Liberties Political_Rights"
572 foreach var of global FH_vars {
573 //Percentile Ranks for LISPs
574 preserve
575 keep if Income == "LISP"
576 PercentRankInc `var' //This program is listed above, and calculates the percentile rank
using MCC's Methodology
577 save "${created_data}\FH `var' LISP.dta", replace
578 restore
579
580 //Percentile Ranks for HISPs
581 preserve
582 keep if Income == "HISP"
583 PercentRankInc `var'
584
585 //Appending other scorecard pool data
586 append using "${created_data}\FH `var' LISP.dta"
587 keep Country percentile Score_In_FY20 Income
588 gen Indicator = "`var'"
589 rename percentile Percentile
590 if "`var'" == "Civil_Liberties" { //We need to use an if statement here because we can't
put a local macro at the beginning of the filename
591 save "${created_data}\Civil_Liberties.dta", replace
592 }
593 else {
594 save "${created_data}\Political_Rights.dta", replace
595 }
596
597 export excel "${final_data}\FY20_`var'.xlsx", firstrow(variables) replace
598 restore
599 }
600
601
602 ///////////////////////////////////////////////////
603 // Education Indicators //
604 ///////////////////////////////////////////////////
605
606 //Download Instructions:
607 /*
608 Go to http://data.uis.unesco.org/
609 On the left panel select Education > Education > Education (full dataset)
610 Click Customise > Selection > Indicator
611 Click Unselect Items > Unselect all then Collapse all
612 Click Indicator > Financial Resources > Government Expenditure on Education >
613 > Government Expenditure on Education as a percentage of GDP
614 > Government Expenditure on Primary Education as a Percentage of GDP (%)
615
616 Then select Completion > Completion and Graduation Ratios >
617 > Gross Intake Ratio to Last Grade > Gross intake ratio to the last grade of primary
education, female (%)
618
619 Then select Participation > Enrollment Ratios >
620 > Gross Enrollment Ratio by Level of Education >
621 > Gross enrollment ratio, lower secondary, female (%)
622
623 Click Time > Select latest data > Select last 50 years
624
625 Click View Data

```

```

626 Go to Export > Text File (CSV) > Download
627
628 Check if columns and sheet names still match
629 Save in the ${raw_data} folder as Education Data YEAR.csv.xls
630 */
631 set more off
632 import delimited "${raw_data}\Education Data 2019.csv", clear
633 drop v6
634 //These countries would be removed when the GNI data is merged anyway, but we need to drop
them early due to the way the reshaping process handles duplicates
635 drop if country == "World" | country == "Latin America and the Caribbean" | country ==
"Small Island Developing States" /*
636 */ | country == "Africa (Northern)" | country == "Africa (Sub-Saharan) " | country == "Arab
States" /*
637 */ | country == "Arab States" | country == "Asia (Central and Southern)" | country == "Asia
(Central)" /*
638 */ | country == "Asia (Eastern and South-eastern)" | country == "Asia (Eastern)" | country
== "Asia (South-eastern)" /*
639 */ | country == "Asia (Southern)" | country == "Asia (Western)" | country == "Africa
(Sub-Saharan)" | country == "East Asia and the Pacific" /*
640 */ | country == "Central Asia" | country == "Central and Eastern Europe" | country == "High
income countries" | country == "Landlocked Developing Countries" /*
641 */ | country == "Least Developed Countries" | country == "Low income countries" | country ==
"Lower middle income countries" | country == "Middle income countries" /*
642 */ | country == "North America and Western Europe" | country == "Northern America" | country
== "Northern America and Europe" | country == "Oceania" /*
643 */ | country == "Oceania (Australia/New Zealand)" | country == "Oceania (excl.
Australia/New Zealand)" | country == "South and West Asia" | country == "Sub-Saharan Africa"
/*
644 */ | country == "Upper middle income countries" | country == "Western Asia and Northern
Africa" | country == "Europe" | country == "Northern America" | country == "Northern
America "
645
646
647
648 //Renaming countries for consistency
649 WBNaming country
650 rename time year
651
652 //Reshaping the data into a "wide" format so it is easier to use
653 **EdExpense
654 preserve
655 keep if indicator == "Government expenditure on primary education as a percentage of GDP (%)"
656 keep country year value
657 reshape wide value, i(country) j(year)
658 save "${created_data}\EdExpense FY20.dta" , replace
659 restore
660
661 **Primary
662 preserve
663 keep if indicator == "Gross intake ratio to the last grade of primary education, female (%)"
664 keep country year value
665 reshape wide value, i(country) j(year)
666 save "${created_data}\Primary FY20.dta" , replace
667 restore
668
669 **Secondary
670 keep if indicator == "Gross enrolment ratio, lower secondary, female (%)"
671 keep country year value
672 reshape wide value, i(country) j(year)
673 save "${created_data}\Secondary FY20.dta" , replace
674
675
676 **Merging
677 *****
678 //For these indicators, MCC takes the most recent data point from the last 7 years of data
(including the current year, 2019)
679 //Other documentation states that we take the last 6 years of data, this is because only
one country has data for 2019
680 //Therefore it is the 6 most recent years of data starting in 2018 or the 7 most recent

```

```

starting in 2019
681 //The following code takes the most recent value for each indicator in the last 7 years
(since 2013)
682 set more off
683 //Ed Expenditures
684 use "${created_data}\EdExpense FY20.dta", clear
685 keep country value2013-value2019
686 egen EdExpense = rowlast(value*) //Using the last (most recent) value, note that the
variables must be sorted chronologically for this to work
687 keep country EdExpense
688 //Primary Completion
689 merge 1:1 country using "${created_data}\Primary FY20.dta" , nogen
690 keep country EdExpense value2013-value2019 //Limiting to the last 7 years
691 egen Primary = rowlast(value*) //Taking the most recent data
692 keep country Primary EdExpense
693 //Secondary Enrollment
694 merge 1:1 country using "${created_data}\Secondary FY20.dta", nogen
695 keep country Primary EdExpense value2013-value2019
696 egen Secondary = rowlast(value*)
697 keep country Secondary EdExpense Primary
698 rename country Country
699 WBNaming Country
700 merge 1:1 Country using "${created_data}\GNI_FY20.dta" , nogen keep(3)
701
702 global EdIndicators "Secondary Primary EdExpense"
703 save "${created_data}\Ed Full.dta", replace
704
705 //The following sections calculate percentile ranks for each of the three education
indicators and generate the files:
706 preserve
707 keep if Income == "LISP"
708 PercentRankInc EdExpense
709 save "${created_data}\Ed EdExpense LISP.dta", replace
710 restore
711 preserve
712 keep if Income == "HISP"
713 PercentRankInc EdExpense
714 append using "${created_data}\Ed EdExpense LISP.dta"
715 keep Country Income percentile Score_In_FY20
716 gen Indicator = "Primary_Education_Expenditures"
717 rename percentile Percentile
718 save "${created_data}\Primary Education Expenditures.dta", replace
719 export excel using "${final_data}\FY20 Primary Education Expenditures.xlsx", firstrow(
variables) replace
720 restore
721
722 preserve
723 keep if Income == "LISP"
724 PercentRankInc Primary
725 keep Country Income percentile Score_In_FY20
726 gen Indicator = "Primary_Girls_Education_Completion"
727 rename percentile Percentile
728 save "${created_data}\Primary Girls Education Completion.dta", replace
729 export excel using "${final_data}\FY20 Primary Girls Education Completion.xlsx", firstrow(
variables) replace
730 restore
731
732 preserve
733 keep if Income == "HISP"
734 PercentRankInc Secondary
735 keep Country Income percentile Score_In_FY20
736 gen Indicator = "Secondary_Girls_Education_Enrollment"
737 rename percentile Percentile
738 save "${created_data}\Secondary Girls Education Enrollment.dta", replace
739 export excel using "${final_data}\FY20 Secondary Girls Education Enrollment.xlsx", firstrow(
variables) replace
740 restore
741
742
743 ///////////////////////////////////////////////////

```

```

744 // Immunization Rate Indicator //
745 ///////////////////////////////////////////////////
746 //Download Instructions
747 /*
748 Download from https://www.who.int/immunization/monitoring_surveillance/data/en/
749 Under 4. Immunization coverage or administered doses, select:
750 4.6 WHO/UNICEF Estimates of National Immunization Coverage (excel)
751
752 Check if columns and sheet names still match (i.e. is the sheet still called MCV1 and the
753 Country variable still called Cname)
754 Save in the ${raw_data} folder as WHO coverage_estimates_series.xls
755 */
756 //This indicator is an average of two measures of immunization: MCV1 and DTP3
757 //The following code performs this calculation and outputs the final file.
758 set more off
759 import excel "${raw_data}\WHO coverage_estimates_series.xls", sheet("MCV1") clear
760
761 //MCV1
762 foreach var of varlist E-AQ { //This loop allows the first row to be renamed as Y{year}
763   (e.g. Y2016) since Stata cannot take numbers as variable names
764   replace `var' = "Y" + `var'[1] in 1
765 }
766 renvars, map (word(@[1],1))
767 drop in 1
768 destring, replace
769 rename Cname Country
770 keep Country Y*
771 WBNaming Country
772 order _all, sequential
773 keep Country Y2018
774 save "${created_data}\Immunization MCV FY20.dta" , replace
775
776 //DTP3
777 import excel "${raw_data}\WHO coverage_estimates_series.xls", sheet("DTP3") clear
778 foreach var of varlist E-AQ { //This loop allows the first row to be renamed as Y{year}
779   (e.g. Y2016) since Stata cannot take numbers as variable names
780   replace `var' = "Y" + `var'[1] in 1
781 }
782 renvars, map (word(@[1],1))
783 drop in 1
784 destring, replace
785 rename Cname Country
786 keep Country Y*
787 WBNaming Country
788 order _all, sequential
789 keep Country Y2018
790 save "${created_data}\Immunization DTP FY20.dta" , replace
791
792 **Merging
793 *****
794 //Merging both files and then merging them with the GNI data
795 set more off
796 use "${created_data}\Immunization MCV FY20.dta", clear
797 keep Country Y2018
798 rename Y2018 MCV1
799 merge 1:1 Country using "${created_data}\Immunization DTP FY20.dta"
800 keep Country Y2018 MCV1
801 rename Y2018 DTP3
802 WBNaming Country
803 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
804
805 egen Score = rowmean(MCV1 DTP3) //Taking the average of the two values
806
807 save "${created_data}\Imm Full.dta", replace
808
809 preserve
810 keep if Income == "LISP"
811 rename Score Immunization Rate LISP
812 PercentRankInc Immunization_Rate_LISP
813 save "${created_data}\Imm LISP.dta", replace

```

```

811
812 restore
813 keep if Income == "HISP"
814 rename Score Immunization_Rate_HISP
815 PercentRankInc Immunization_Rate_HISP
816 append using "${created_data}\Imm LISP.dta"
817 keep Country Income percentile Score_In_FY20
818 gen Indicator = "Immunization_Rate"
819 rename percentile Percentile
820 save "${created_data}\Immunization_Rate.dta", replace
821 export excel using "${final_data}\FY20_Immunization_Rate.xlsx", firstrow(variables) replace
822
823
824 ///////////////////////////////////////////////////
825 // Health Expenditures //
826 ///////////////////////////////////////////////////
827
828 //Download Instructions
829 /*
830 Go to http://apps.who.int/nha/database/Select/Indicators/en
831 Select Indicators > Financing Sources >
832 Domestic General Government Health Expenditure (GGHE-D) as % of Gross Domestic Product (GDP)
833 Click Next, select all countries, click next
834 Select all years, click next
835 Select % Gross domestic product (GDP)
836 click view data and build report
837 Click Excel Workbook
838
839 Save as NHA indicators.xlsx in the raw data folder
840 check to see if the columns and rows still match.
841 */
842 set more off
843 import excel "${raw_data}\NHA indicators.xlsx", sheet("Table") clear
844 drop B C
845 drop in 2
846
847 foreach var of varlist D-T { //replacing the first row with values that have characters in
848 them so that variables can be renamed using renvars
849 replace `var' = "Y" + `var'[1] in 1
850 }
851 renvars, map (word(@[1],1))
852 drop in 1
853 rename Countries Country
854 WBNaming Country
855 destring, replace
856 save "${created_data}\Health FY20.dta", replace
857
858 **Merging
859 *****
860 keep Country Y2016 //Keep the most recent year
861 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
862 save "${created_data}\Health Full.dta", replace
863
864 preserve
865 keep if Income == "LISP"
866 PercentRankInc Y2016
867 save "${created_data}\Health LISP.dta", replace
868 restore
869 keep if Income == "HISP"
870 PercentRankInc Y2016
871 save "${created_data}\Health HISP.dta", replace
872 append using "${created_data}\Health LISP.dta"
873
874 keep Country Income percentile Score_In_FY20
875 gen Indicator = "Health_Expenditures"
876 rename percentile Percentile
877 save "${created_data}\Health Expenditures.dta", replace
878 export excel using "${final_data}\FY20 Health Expenditures.xlsx", firstrow(variables) replace
879

```

```

880 ///////////////////////////////////////////////////
881 // Gender in the Economy //
882 ///////////////////////////////////////////////////
883
884
885 //Download Instructions:
886 /*
887 Go to Women, Business and the Law. Go to their reports page:
888 https://wbl.worldbank.org/en/reports
889 Find the most recent report with new data, and click on data
890 Save as the file referenced below in the raw data folder
891 */
892
893 set more off
894 import excel "${raw_data}\wblrawdata2010201829march2018.xlsx", sheet("WBL2018") clear
895 set more off
896 tostring all, replace
897 replace AT = "Can an unmarried woman be HoH like an unmarried man?" in 1
898 replace AU = "Can a married woman be HoH like a married man?" in 1
899
900 //The following loop cleans up the first row into variable names that are not too long for
901 Stata to handle
902
903 foreach var of varlist _all {
904   if "`='var'[1]'" == "." {
905     drop `var'
906   }
907   else {
908     //drop `var' if `var' == "."
909     replace `var' = subinstr("`='var'[1]'", `""', "",.) in 1
910     replace `var' = subinstr("`='var'[1]'", "If customary law is recognized as a valid source
of law under the constitution, is it invalid if it violates constitutional provisions on
nondiscrimination or equality?", "If vld cust law, does const invldate if disc?") in 1
911     replace `var' = subinstr("`='var'[1]'", "If personal law is recognized as a valid source of
law under the constitution, is it invalid if it violates constitutional provisions on
nondiscrimination or equality?", "If vld personal law, does const invldate if disc?") in 1
912
913     replace `var' = subinstr("`='var'[1]'", " (reserved seats) in place", "",.) in 1
914     replace `var' = subinstr("`='var'[1]'", "legislative", "Lgs.",.) in 1
915     replace `var' = subinstr("`='var'[1]'", "sanctions for noncompliance", "sanc. for ncomp.",.)
in 1
916     replace `var' = subinstr("`='var'[1]'", "with mandated quotas", "w/quotas",.) in 1
917     replace `var' = subinstr("`='var'[1]'", "(e.g. financial)", "($)",.) in 1
918     replace `var' = subinstr("`='var'[1]'", "political parties", "pol. part.",.) in 1
919     replace `var' = subinstr("`='var'[1]'", "women on candidate lists", "w. cand.",.) in 1
920     replace `var' = subinstr("`='var'[1]'", "nondiscrimination", "ndiscrm",.) in 1
921     replace `var' = subinstr("`='var'[1]'", "constitution", "const",.) in 1
922     replace `var' = subinstr("`='var'[1]'", "customary", "cust",.) in 1
923     replace `var' = subinstr("`='var'[1]'", "explicitly", "",.) in 1
924     replace `var' = subinstr("`='var'[1]'", "recognized", "recg",.) in 1
925     replace `var' = subinstr("`='var'[1]'", "valid", "vld",.) in 1
926     replace `var' = subinstr("`='var'[1]'", "explicitly", "",.) in 1
927     replace `var' = subinstr("`='var'[1]'", "unmarried", "umar",.) in 1
928     replace `var' = subinstr("`='var'[1]'", "married", "mar",.) in 1
929     replace `var' = subinstr("`='var'[1]'", " or pursue a trade or profession", "",.) in 1
930     replace `var' = subinstr("`='var'[1]'", "children", "kids",.) in 1
931     replace `var' = subinstr("`='var'[1]'", "non-national", "",.) in 1
932     replace `var' = subinstr("`='var'[1]'", "legal", "leg",.) in 1
933     replace `var' = subinstr("`='var'[1]'", "responsibility", "resp",.) in 1
934     replace `var' = subinstr("`='var'[1]'", "financially", "fin",.) in 1
935     replace `var' = subinstr("`='var'[1]'", "husband", "hus",.) in 1
936     replace `var' = subinstr("`='var'[1]'", "administers", "admns",.) in 1
937     replace `var' = subinstr("`='var'[1]'", "property", "prop",.) in 1
938     replace `var' = subinstr("`='var'[1]'", "formal", "frml",.) in 1
939     replace `var' = subinstr("`='var'[1]'", "their decisions be appealed", "they be appealed",.)
in 1
940     replace `var' = subinstr("`='var'[1]'", "recognize", "recog",.) in 1
941     replace `var' = subinstr("`='var'[1]'", "personal", "prsnl",.) in 1

```



```

942 replace `var' = subinstr("`var'[1]", "court", "crt",.) in 1
943 replace `var' = subinstr("`var'[1]", "private credit bureau", "priv cred bur",.) in 1
944 replace `var' = subinstr("`var'[1]", "public credit registry", "pub cred reg",.) in 1
945 replace `var' = subinstr("`var'[1]", "public credit registries", "pub cred regs",.) in 1
946 replace `var' = subinstr("`var'[1]", "information", "info",.) in 1
947 replace `var' = subinstr("`var'[1]", "retailers", "rtlrs",.) in 1
948 replace `var' = subinstr("`var'[1]", "microfinance", "microfin",.) in 1
949 replace `var' = subinstr("`var'[1]", "institutions", "insts",.) in 1
950 replace `var' = subinstr("`var'[1]", "institution", "inst",.) in 1
951 replace `var' = subinstr("`var'[1]", "discrimination", "disc",.) in 1
952 replace `var' = subinstr("`var'[1]", "on the basis of", "on",.) in 1
953 replace `var' = subinstr("`var'[1]", "nonpregnant and nonnursing", "npreg/nonurse",.) in 1
954 replace `var' = subinstr("`var'[1]", "protection", "prtctn",.) in 1
955 replace `var' = subinstr("`var'[1]", "distance", "dist",.) in 1
956 replace `var' = subinstr("`var'[1]", "maintaining", "maintng",.) in 1
957 replace `var' = subinstr("`var'[1]", "cover prohibition", "prhbt",.) in 1
958 replace `var' = subinstr("`var'[1]", "authorizing or knowingly entering into",
"authrznng/knwinly marrying",.) in 1
959 replace `var' = subinstr("`var'[1]", "penalties in the law", "penalties",.) in 1
960
961 label var `var' "`var'[1]"
962
963 replace `var' = subinstr("`var'[1]", " ", "_",.) in 1
964 replace `var' = subinstr("`var'[1]", "/", "_",.) in 1
965 replace `var' = subinstr("`var'[1]", ",", "_",.) in 1
966 replace `var' = subinstr("`var'[1]", ".", "_",.) in 1
967 replace `var' = subinstr("`var'[1]", "?", "_",.) in 1
968 replace `var' = subinstr("`var'[1]", "(", "_",.) in 1
969 replace `var' = subinstr("`var'[1]", ")", "_",.) in 1
970 replace `var' = subinstr("`var'[1]", "`", "_",.) in 1
971 replace `var' = subinstr("`var'[1]", "'", "_",.) in 1
972 replace `var' = subinstr("`var'[1]", "`", "_",.) in 1
973 replace `var' = subinstr("`var'[1]", "-", "_",.) in 1
974
975
976 replace `var' = subinstr("`var'[1]", "Does_the_", "",.) in 1
977 replace `var' = subinstr("`var'[1]", "What are the ", "",.) in 1
978 replace `var' = subinstr("`var'[1]", "women representatives", "w_cncls",.) in 1
979 replace `var' = subinstr("`var'[1]", "in elections for_", "",.) in 1
980 replace `var' = subinstr("`var'[1]", "Are there_", "",.) in 1
981 replace `var' = subinstr("`var'[1]", "for_", "",.) in 1
982 replace `var' = subinstr("`var'[1]", "Can_an_", "",.) in 1
983 replace `var' = subinstr("`var'[1]", "Can a ", "",.) in 1
984 replace `var' = subinstr("`var'[1]", "citizenship", "ctzsh",.) in 1
985 replace `var' = subinstr("`var'[1]", "confer", "cfr",.) in 1
986 replace `var' = subinstr("`var'[1]", "prohibit", "prbt",.) in 1
987 replace `var' = subinstr("`var'[1]", "creditor", "crdr",.) in 1
988 replace `var' = subinstr("`var'[1]", "What percentage", "perc",.) in 1
989 replace `var' = subinstr("`var'[1]", "wages", "wgs",.) in 1
990 replace `var' = subinstr("`var'[1]", "paid", "pd",.) in 1
991 replace `var' = subinstr("`var'[1]", "How many_", "",.) in 1
992 replace `var' = subinstr("`var'[1]", "parental", "prt1",.) in 1
993 replace `var' = subinstr("`var'[1]", "leave", "lv",.) in 1
994 replace `var' = subinstr("`var'[1]", "must be taken by the_", "mst_tak",.) in 1
995 replace `var' = subinstr("`var'[1]", "mandate", "mdt",.) in 1
996 replace `var' = subinstr("`var'[1]", "based on", "on",.) in 1
997 replace `var' = subinstr("`var'[1]", "What is the age", "age",.) in 1
998 replace `var' = subinstr("`var'[1]", "at which a_", "",.) in 1
999 replace `var' = subinstr("`var'[1]", "and recieve full", "full",.) in 1
1000 replace `var' = subinstr("`var'[1]", "and recieve partial", "partial",.) in 1
1001 replace `var' = subinstr("`var'[1]", "can retire", "retire",.) in 1
1002 replace `var' = subinstr("`var'[1]", "What is the mandatory ", "mndt",.) in 1
1003 replace `var' = subinstr("`var'[1]", "retirement", "retrmt",.) in 1
1004 replace `var' = subinstr("`var'[1]", "Can_npregnonurse_women", "nprg_nse_w",.) in 1
1005 replace `var' = subinstr("`var'[1]", "Does_domestic_violence_legislation", "dom_vio_leg",
,.) in 1
1006 replace `var' = subinstr("`var'[1]", "legislation", "leg",.) in 1
1007 replace `var' = subinstr("`var'[1]", "Is there ", "",.) in 1
1008 replace `var' = subinstr("`var'[1]", "sexual_harassment", "sx_hrsmt",.) in 1
1009 replace `var' = subinstr("`var'[1]", "What_is_the_leg", "leg",.) in 1

```

```

1010 replace `var' = subinstr("`='var'[1]'", "What is the minimum", "min",.) in 1
1011 replace `var' = subinstr("`='var'[1]'", "marriage", "marr",.) in 1
1012 replace `var' = subinstr("`='var'[1]'", "with", "w",.) in 1
1013 replace `var' = subinstr("`='var'[1]'", "judicial", "jud",.) in 1
1014 replace `var' = subinstr("`='var'[1]'", "authorization", "authztn",.) in 1
1015 replace `var' = subinstr("`='var'[1]'", "_", "",.) in 1
1016 replace `var' = subinstr("`='var'[1]'", "_", "_",.) in 1
1017 replace `var' = abbrev(`var', 31) in 1
1018 replace `var' = subinstr("`='var'[1]'", "~", "",.) in 1
1019 }
1020 }
1021
1022 renvars, map (word(@[1],1))
1023
1024 //The following section keeps only the variables that MCC uses to construct the Gender in
the Economy Indicator.
1025 keep Economy /*
1026 Accessing Institutions
1027 /* umar_woman_obtain_a_national__ mar_woman_obtain_a_national Ic /*
1028 /* umar_woman_travel_outside_theo mar_woman_travel_outside_the_u /*
1029 /* umar_woman_travel_outside_hero mar_woman_travel_outside_hero m /*
1030 /* umar_woman_get_a_job_in_the_se mar_woman_get_a_job_in_the_sa /*
1031 /* umar_woman_sign_a_contract_inh mar_woman_sign_a_contract_in_e /*
1032 /* umar_woman_register_a_businesi mar_woman_register_a_businessn /*
1033 /* umar_woman_open_a_bank_accounti mar_woman_open_a_bank_accountn /*
1034 /* umar_woman_choose_where_to_li mar_woman_choose_where_to_livi /*
1035 /* umar_woman_cfr_ctzsh_to_her_ks mar_woman_cfr_ctzsh_to_her_ki /*
1036 /* umar_woman_be_HoH_like_an_umam mar_woman_be_HoH_like_a_mar_man /*
1037 /* If vld cust law does const ind Do_mar_couples_jointly_share_g /*
1038 Using Property
1039 /* law_provide_the_valuation_of_n Who_legly_adms_marital_prop /*
1040 /* Do_umar_men_and_umar_women_ha Do_mar_men_and_mar_women_haveq /*
1041 /* Do_sons_and_daughters_have_eql Do_female_and_male_surviving_o /*
1042 Going to Court
1043 /* Does_a_womans_testimony_carryh /*
1044 Getting a Job
1045 /* nprg_nse_w_work_the_same_nighh nprg_nse_w_do_the_same_jobs_am /*
1046 Protecting Women From Violence
1047 /* domestic_violence_leg clear_criminal_penalties_domei /*
1048 /* a_specialized_crt_or_procedurc leg_that_specifically_address /*
1049 /* criminal_penalties_sx_hrsmt_ie leg_age_of_marr_girls /*
1050 /* any_exceptions_to_the_leg_agef law_prbt_or_invldate_child_ora /*
1051 /* penalties_authrzngknwinly_mari
1052 rename Economy Country
1053
1054 drop in 1
1055 destring, replace
1056 //Some indicators count for half a point, others count for a full point. The following
global macros split the variables above into these two categories
1057 global halfpoint_indicators "umar_woman_obtain_a_national__
mar_woman_obtain_a_national_Ic umar_woman_travel_outside_theo
mar_woman_travel_outside_the_u umar_woman_travel_outside_hero
mar_woman_travel_outside_hero_m umar_woman_get_a_job_in_the_se
mar_woman_get_a_job_in_the_sa umar_woman_sign_a_contract_inh
mar_woman_sign_a_contract_in_e umar_woman_register_a_businesi
mar_woman_register_a_businessn umar_woman_open_a_bank_accounti
mar_woman_open_a_bank_accountn umar_woman_choose_where_to_li
mar_woman_choose_where_to_livi umar_woman_cfr_ctzsh_to_her_ks
mar_woman_cfr_ctzsh_to_her_ki umar_woman_be_HoH_like_an_umam
mar_woman_be_HoH_like_a_mar_man Do_umar_men_and_umar_women_ha_
Do_mar_men_and_mar_women_haveq "
1058 global fullpoint_indicators "If vld cust law does const ind
Do_mar_couples_jointly_share_g law_provide_the_valuation_of_n
Who_legly_adms_marital_prop Do_sons_and_daughters_have_eql
Do_female_and_male_surviving_o Does_a_womans_testimony_carryh
nprg_nse_w_work_the_same_nighh nprg_nse_w_do_the_same_jobs_am domestic_violence_leg
clear_criminal_penalties_domei a_specialized_crt_or_procedurc
leg_that_specifically_address criminal_penalties_sx_hrsmt_ie
law_prbt_or_invldate_child_ora penalties_authrzngknwinly_mari"
1059

```

```

1060 //The following assigns 0.5 points to any halfpoint indicator with a "No" or ".." value
1061 foreach halfpoint of global halfpoint_indicators {
1062 replace `halfpoint' = "0.5" if `halfpoint' == "No"
1063 replace `halfpoint' = "0" if `halfpoint' == "Yes"
1064 replace `halfpoint' = "0" if `halfpoint' == "yes"
1065 replace `halfpoint' = "." if `halfpoint' == "N/A"
1066 replace `halfpoint' = "0.5" if `halfpoint' == ".."
1067 }
1068 destring, replace
1069
1070 //The following assigns 1 point to answers of "No", "Husband", "Wife", or ".."
1071 foreach fullpoint of global fullpoint_indicators {
1072 replace `fullpoint' = "1" if `fullpoint' == "No"
1073 replace `fullpoint' = "0" if `fullpoint' == "Yes"
1074 replace `fullpoint' = "0" if `fullpoint' == "yes"
1075 replace `fullpoint' = "1" if `fullpoint' == "Husband"
1076 replace `fullpoint' = "1" if `fullpoint' == "husband"
1077 replace `fullpoint' = "1" if `fullpoint' == "Wife"
1078 replace `fullpoint' = "1" if `fullpoint' == "wife"
1079 replace `fullpoint' = "0" if `fullpoint' == "Other"
1080 replace `fullpoint' = "0" if `fullpoint' == "other"
1081 replace `fullpoint' = "0" if `fullpoint' == "Both must agree"
1082 replace `fullpoint' = "0" if `fullpoint' == "both must agree"
1083 replace `fullpoint' = "0" if `fullpoint' == "Original owner"
1084 replace `fullpoint' = "0" if `fullpoint' == "original owner"
1085 replace `fullpoint' = "0" if `fullpoint' == "Original Owner"
1086 replace `fullpoint' = "0" if `fullpoint' == "Separate with spousal consent"
1087 replace `fullpoint' = "0" if `fullpoint' == "separate with spousal consent"
1088 replace `fullpoint' = "." if `fullpoint' == "N/A"
1089 replace `fullpoint' = "1" if `fullpoint' == ".."
1090 }
1091
1092 //The following assigns 1 point to the indicator for answers of "..", "Yes", or values
1093 //under 18 Note that these indicators are reversed from other indicators in the loops above
1094 replace leg_age_of_marr_girls = "1" if leg_age_of_marr_girls == ".."
1095 replace any_exceptions_to_the_leg_agef = "1" if any_exceptions_to_the_leg_agef == "Yes"
1096 replace any_exceptions_to_the_leg_agef = "1" if any_exceptions_to_the_leg_agef == "yes"
1097 replace any_exceptions_to_the_leg_agef = "0" if any_exceptions_to_the_leg_agef == "No"
1098 replace any_exceptions_to_the_leg_agef = "0" if any_exceptions_to_the_leg_agef == "no"
1099 destring, replace
1100 replace leg_age_of_marr_girls = 1 if leg_age_of_marr_girls < 18
1101 replace leg_age_of_marr_girls = 0 if leg_age_of_marr_girls >= 18
1102
1103 //The following creates a macro of all of the relevant variables, then sums those across.
1104 unab Allvars : _all
1105 unab exclude : Country
1106 local Addlist : list Allvars - exclude
1107 egen Gender_score = rowtotal(`Addlist')
1108 label variable Gender_score "2019"
1109 keep Country Gender_score
1110 WBNaming Country
1111 save "${created_data}\Gender FY20.dta", replace
1112
1113 **Merging
1114 *****
1115
1116 use "${created_data}\Gender FY20.dta", clear
1117 keep Country Gender_score
1118 WBNaming Country
1119 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1120
1121 save "${created_data}\Gender Full.dta", replace
1122 preserve
1123 keep if Income == "LISP"
1124 PercentRankInc Gender_score
1125 replace percentile = 1-percentile //Note this is different from other indicators due to the
//inverse scale
1126 save "${created_data}\Gender LISP.dta", replace
1127 restore

```

```

1128 keep if Income == "HISP"
1129 PercentRankInc Gender_score
1130 replace percentile = 1-percentile //Note this is different from other indicators due to the
inverse scale
1131 save "${created_data}\Gender HISP.dta", replace
1132 append using "${created_data}\Gender LISP.dta"
1133
1134
1135 keep Country Income percentile Score_In_FY20
1136 gen Indicator = "Gender_In_The_Economy"
1137 rename percentile Percentile
1138 save "${created_data}\Gender In The Economy.dta", replace
1139 export excel using "${final_data}\FY20_Gender_In_The_Economy.xlsx", firstrow(variables)
replace
1140
1141 ////////////////////////////////////////////////////
1142 // Natural Resource Protection //
1143 ////////////////////////////////////////////////////
1144
1145 //Download Instructions:
1146 /*
1147 Go to CIESIN's Natural Resource Management Index
1148 https://sedac.ciesin.columbia.edu/data/collection/nrmi/sets/browse
1149 Click on Data Sets, and select the most recent release of the Natural resource protection
and Child Health Indicators
1149 Click on Data Download. You may need to log into NASA's website to view the data.
1150 Select the zip release
1151 Save as the file referenced below in the raw data folder
1152 */
1153
1154
1155 import excel "${raw_data}\NRPI.xlsx", sheet("NRPI_proximity") firstrow clear
1156 replace NRPI_v2019_19 = 100 if CountryName == "Morocco" //Morocco's value was incorrectly
entered as 99.99999999999999 but should be 100.
1157 WBNaming CountryName
1158 label var CountryName "Country"
1159 keep CountryName NRPI*
1160 save "${created_data}\NRP FY20.dta", replace
1161
1162 **Merging
1163 *****
1164
1165 use "${created_data}\NRP FY20.dta", clear
1166 rename CountryName Country
1167 keep Country NRPI_v2019_19
1168 WBNaming Country
1169 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1170 save "${created_data}\NRP Full.dta", replace
1171 preserve
1172 keep if Income == "LISP"
1173 PercentRankInc NRPI_v2019_19
1174 save "${created_data}\NRP LISP.dta", replace
1175 restore
1176 keep if Income == "HISP"
1177 PercentRankInc NRPI_v2019_19
1178 save "${created_data}\NRP HISP.dta", replace
1179 append using "${created_data}\NRP LISP.dta"
1180
1181
1182 keep Country Income percentile Score_In_FY20
1183 gen Indicator = "Natural_Resource_Protection"
1184 rename percentile Percentile
1185 save "${created_data}\Natural Resource Protection.dta", replace
1186 export excel using "${final_data}\FY20_Natural_Resource_Protection.xlsx", firstrow(variables
) replace
1187
1188 ////////////////////////////////////////////////////
1189 // Child Health //
1190 ////////////////////////////////////////////////////
1191 //Download Instructions:

```

```

1192 //This should be contained in the same download as the NRP data above
1193
1194 import excel "${raw_data}\CHI_2019.xlsx", sheet("CHI_2019_v3") firstrow clear
1195 keep Country CHI_v2019_18 CHI_v2019_17 CHI_v2019_16 CHI_v2019_15 CHI_v2019_14 CHI_v2019_13
    CHI_v2019_12 CHI_v2019_11 CHI_v2019_10
1196 WBNaming Country
1197
1198 save "${created_data}\Child FY20.dta", replace
1199
1200 **Merging
1201 *****
1202 merge 1:1 Country using "${created_data}\GNI FY20.dta", nogen keep(3)
1203
1204 save "${created_data}\Child Full.dta", replace
1205 preserve
1206 keep if Income == "LISP"
1207 PercentRankInc CHI_v2019_18
1208 save "${created_data}\Child LISP.dta", replace
1209 restore
1210 keep if Income == "HISP"
1211 PercentRankInc CHI_v2019_18
1212 save "${created_data}\Child HISP.dta", replace
1213 append using "${created_data}\Child LISP.dta"
1214
1215 keep Country Income percentile Score_In_FY20
1216 gen Indicator = "Child_Health"
1217 rename percentile Percentile
1218 save "${created_data}\Child Health.dta", replace
1219 export excel using "${final_data}\FY20_Child_Health.xlsx", firstrow(variables) replace
1220
1221
1222 ////////////////////////////////////////////////////////////////////
1223 // Land Rights and Access (IFAD) //
1224 ////////////////////////////////////////////////////////////////////
1225
1226 //Download Instructions
1227 /*
1228 Go to https://webapps.ifad.org/members/gc
1229 click on "All Sessions" on the Right to go here https://webapps.ifad.org/members/gc/sessions
1230 Find the most recent Session, and look for "Programme of work and budget of IFAD and its
    Office of Evaluation for YEAR"
1231 If this is not present, check a previous Session
1232 Scroll to the bottom of this document to the section titled "YEAR Rural Sector Performance
    Assessment (RSPA) scores"
1233 Find the indicator for Access to Land (3.3 in 2018)
1234 Copy all the numbers into a txt file, replace all spaces with commas. Save as a .csv file
1235 Open CSV in excel, the numbers should be in different columns.
1236 Copy the data, use special paste to reformat it so that it is in a single column.
1237 Copy the country labels in one group at a time. Make sure you use the same order as the
    original data.
1238 (You may need to open the file in Adobe Acrobat instead of a web browser to do this).
1239 Delete any cells where the country title is on more than one line. Shift Cells up.
1240 Check to make sure that all of the country data is lined up correctly.
1241 Label the columns "Country" and "Data", Label the sheet IFAD Data
1242
1243 Save as a Excel file (not a CSV) named IFAD Data.xlsx
1244
1245
1246 If data is not available in the current year for a country, MCC pulls forward data from the
    previous report (back to 2014 for FY20).
1247 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
1248 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"
1249 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
1250 Label the columns "Country" and "Data", label the sheet IFAD Data
1251 Save as IFAD_2015.xlsx

```

```

1252
1253 Repeat this process, for the 2015 session (38th) of the governing council. These numbers
      should be from the 2014 RSPA.
1254 Save as IFAD_2014.xlsx
1255
1256 */
1257
1258 set more off
1259 import excel "${raw_data}\IFAD Data.xlsx", clear firstrow sheet("IFAD Data")
1260 WBNaming Country
1261 rename Data Y2020
1262 save "${created_data}\IFAD Data.dta", replace
1263
1264 import excel "${raw_data}\IFAD_2015.xlsx", clear firstrow sheet("IFAD Data")
1265 WBNaming Country
1266 rename Data Y2015
1267 merge 1:1 Country using "${created_data}\IFAD_Data.dta", nogen
1268 replace Y2020 = Y2015 if missing(Y2020)
1269 save "${created_data}\IFAD_Data.dta", replace
1270
1271 import excel "${raw_data}\IFAD_2014.xlsx", clear firstrow sheet("IFAD Data")
1272 WBNaming Country
1273 rename Data Y2014
1274 merge 1:1 Country using "${created_data}\IFAD_Data.dta", nogen
1275 replace Y2020 = Y2014 if missing(Y2020)
1276
1277
1278 egen Max_Score2020 = max(Y2020)
1279 egen Min_Score2020 = min(Y2020)
1280 gen IFAD_Score2020 = 1-((Max_Score2020-Y2020)/(Max_Score2020-Min_Score2020))
1281
1282 //keep Country IFAD_Score*
1283 save "${created_data}\IFAD_Score_Normalized.dta", replace
1284
1285
1286 //This indicator is only half of the Land Rights and Access Indicator
1287 //We need to add in the Doing Business Registering Property data in order to create the
      full indicator.
1288
1289
1290
1291 ///////////////////////////////////////////////////
1292 // Doing Business //
1293 ///////////////////////////////////////////////////
1294
1295 //Download Instructions:
1296 /*
1297 Go to https://www.doingbusiness.org/en/data and select historical data create a custom
      dataset
1298 This should take you to: https://www.doingbusiness.org/en/custom-query
1299 Choose all Economies
1300 Choose Topics "Starting a Business" "Getting Credit" and "Registering Property"
1301 Select years back to 2015
1302 Click Create report
1303 Click the excel button to download a report
1304 Save as the file below
1305 */
1306
1307
1308 import excel "${raw_data}\Doing Business 2020.xlsx", clear
1309 //The following changes variable names that are too long for Stata
1310 foreach var of varlist all {
1311 label var `var' "`='var'[1]'"
1312 replace `var' = subinstr("`='var'[1]'", " ", "_",.) in 1
1313 replace `var' = subinstr("`='var'[1]'", ")", "_",.) in 1
1314 replace `var' = subinstr("`='var'[1]'", "(", "_",.) in 1
1315 replace `var' = subinstr("`='var'[1]'", "-", "_",.) in 1
1316 replace `var' = subinstr("`='var'[1]'", "%", "pc",.) in 1
1317 replace `var' = subinstr("`='var'[1]'", "?", "_",.) in 1
1318 replace `var' = subinstr("`='var'[1]'", "__", "_",.) in 1

```

```

1319 replace `var' = substr("`var'[1]", "_", ".",) in 1
1320 replace `var' = substr("`var'[1]", "doing_business", "do_biz",.) in 1
1321 replace `var' = substr("`var'[1]", "Registering_property_", "Rg_prp_",.) in 1
1322 replace `var' = substr("`var'[1]", "Registering_Property_", "Rg_Prp_",.) in 1
1323 replace `var' = substr("`var'[1]", "Quality_of_land_administration", "Ql_ld_adm",.) in 1
1324 replace `var' = substr("`var'[1]", "Starting_a_Business", "Start_Biz",.) in 1
1325 replace `var' = substr("`var'[1]", "Getting_Credit", "Get_Cr",.) in 1
1326 replace `var' = substr("`var'[1]", "Strength_of", "St",.) in 1
1327 replace `var' = substr("`var'[1]", "Depth_of", "Dp",.) in 1
1328 replace `var' = substr("`var'[1]", "index", "",.) in 1
1329 replace `var' = substr("`var'[1]", "_", ".",) in 1
1330 replace `var' = substr("`var'[1]", " ", " ",.) in 1
1331 replace `var' = abbrev(`var', 27) in 1
1332 replace `var' = substr("`var'[1]", "~", "",.) in 1
1333 //Coding "No Practice" as "-999999999", this will be changed back to a real value below
1334 replace `var' = "-999999999" if `var' == "No Practice"
1335 }
1336 renvars, map (word(@[1],1))
1337 drop in 1
1338
1339 //Keeping only the necessary variables and observations
1340 replace Year = substr(Year,"DB","",.)
1341 destrng, replace
1342 drop if Year < 2015
1343 drop if missing(Year) & Economy == ""
1344
1345 //Setting variable names up so that they can be reshaped, and reshaping the data to a wide
format
1346 unab Allvars : _all
1347 unab exclude : Economy Year
1348 local Addlist1 : list Allvars - exclude
1349
1350 foreach var of varlist `Addlist1' {
1351 rename `var' `var'_
1352 local `var'_lb : variable label `var'
1353 }
1354
1355 unab Allvars : _all
1356 unab exclude : Economy Year
1357 local Addlist2 : list Allvars - exclude
1358 reshape wide `Addlist2' , i(Economy) j(Year)
1359
1360 //Labeling Variables
1361 foreach var of local Addlist1 {
1362 forvalues k = 2015(1)2020 {
1363 label var `var'_`k' "`var'_lb' `k'"
1364 }
1365 }
1366
1367 //Generating a maximum for each variable, and putting it in place of the "No Practice" values
1368 unab Allvars : _all
1369 unab exclude : Economy
1370 local Addlist3 : list Allvars - exclude
1371
1372 foreach var of varlist `Addlist3' {
1373 egen x`var' = max(`var')
1374 replace `var' = x`var' if `var' == -999999999
1375 drop x`var'
1376 }
1377
1378 //Merging
1379 WBNaming Economy
1380 rename Economy Country
1381 label var Country "Country"
1382 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1383 drop GNI
1384 save "${created_data}\All_Doing_Business_Formatted.dta" , replace
1385
1386 //This file will be used in what follows to create three indicators: Business Start Up,
Access to Credit, and (with IFAD) Land Rights and Access.

```

```

1387
1388
1389 ////////////////////////////////////////////////////
1390 // Access to Credit //
1391 ////////////////////////////////////////////////////
1392 use "${created_data}\All_Doing_Business_Formatted.dta", clear
1393 keep Country Get_Cr_St_legal_rights_152_20* Get_Cr_Dp_credit_inform08_20*
1394 //The following computes access to credit scores for a given country by normalizing both
components and then taking an average.
1395 forvalues k=15(1)20 {
1396 gen Norm_Depth_`k' = Get_Cr_Dp_credit_inform08_20`k' * 12
1397 gen Norm_Legal_`k' = Get_Cr_St_legal_rights_152_20`k' * 8
1398 gen Access_score_`k' = (Norm_Depth_`k' + Norm_Legal_`k')/2
1399 }
1400 save "${created_data}\Access_Calculated.dta" , replace
1401
1402
1403 **Merging
1404 *****
1405
1406 use "${created_data}\Access_Calculated.dta", clear
1407 keep Country Access_score_20
1408 rename Access_score_20 Access_score
1409 WBNaming Country
1410 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1411
1412 save "${created_data}\Access_Full.dta", replace
1413 preserve
1414 keep if Income == "LISP"
1415 PercentRankInc Access_score
1416 save "${created_data}\Access_LISP.dta", replace
1417 restore
1418 keep if Income == "HISP"
1419 PercentRankInc Access_score
1420 save "${created_data}\Access_HISP.dta", replace
1421 append using "${created_data}\Access_LISP.dta"
1422
1423 keep Country Income percentile Score_In_FY20
1424 gen Indicator = "Access_To_Credit"
1425 rename percentile Percentile
1426 save "${created_data}\Access_To_Credit.dta", replace
1427 export excel using "${final_data}\FY20_Access_To_Credit.xlsx", firstrow(variables) replace
1428
1429
1430 ////////////////////////////////////////////////////
1431 // Business Start-Up //
1432 ////////////////////////////////////////////////////
1433
1434
1435 use "${created_data}\All_Doing_Business_Formatted.dta", clear
1436 keep Country Start_Biz_Cost_Men_pc_of__20* Start_Biz_Cost_Women_pc_om_20*
Start_Biz_Time_Men_days_20* Start_Biz_Time_Women_days_20*
1437 forvalues k= 15(1)20 {
1438 egen Time_Average`k' = rowmean(Start_Biz_Time_Men_days_20`k' Start_Biz_Time_Women_days_20
`k') //Taking an average of time for men and women
1439 egen Cost_Average`k' = rowmean(Start_Biz_Cost_Men_pc_of__20`k'
Start_Biz_Cost_Women_pc_om_20`k') //Taking an average of cost for men and women
1440 //The following finds the min and max of these variables
1441 egen Max_Time`k' = max(Time_Average`k')
1442 egen Max_Cost`k' = max(Cost_Average`k')
1443 egen Min_Time`k' = min(Time_Average`k')
1444 egen Min_Cost`k' = min(Cost_Average`k')
1445 //The following normalizes these variables to a common scale using the min and max
1446 gen Time_Score`k' = (Max_Time`k'-Time_Average`k')/(Max_Time`k'-Min_Time`k')
1447 gen Cost_Score`k' = (Max_Cost`k'-Cost_Average`k')/(Max_Cost`k'-Min_Cost`k')
1448 gen Business_score_`k' = (Time_Score`k' + Cost_Score`k')/2
1449 }
1450 keep Country Time_Score* Cost_Score* Business_score*
1451 save "${created_data}\Business_Calculated.dta" , replace
1452

```



```

1453  **Merging
1454  *****
1455  use "${created_data}\Business Calculated.dta", clear
1456  keep Country Business_score_20
1457  rename Business_score_20 Business_score
1458  WBNaming Country
1459  merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1460
1461  save "${created_data}\Business Full.dta", replace
1462  preserve
1463  keep if Income == "LISP"
1464  PercentRankInc Business_score
1465  save "${created_data}\Business LISP.dta", replace
1466  restore
1467  keep if Income == "HISP"
1468  PercentRankInc Business_score
1469  save "${created_data}\Business HISP.dta", replace
1470  append using "${created_data}\Business LISP.dta"
1471
1472  keep Country Income percentile Score_In_FY20
1473  gen Indicator = "Business_Start_Up"
1474  rename percentile Percentile
1475  save "${created_data}\Business_Start_Up.dta", replace
1476  export excel using "${final_data}\FY20_Business_Start_Up.xlsx", firstrow(variables) replace
1477
1478
1479  //////////////////////////////////////
1480  //  Land Rights and Access (WB)  //
1481  //////////////////////////////////////
1482
1483  set more off
1484  use "${created_data}\All_Doing_Business_Formatted.dta", clear
1485  keep Country Rg_Prp_Time_days_20* Rg_Prp_Cost_pc_of_properte_20*
1486
1487  forvalues k= 15(1)20 {
1488  egen Max_Time`k' = max(Rg_Prp_Time_days_20`k')
1489  egen Max_Cost`k' = max(Rg_Prp_Cost_pc_of_properte_20`k')
1490  egen Min_Time`k' = min(Rg_Prp_Time_days_20`k')
1491  egen Min_Cost`k' = min(Rg_Prp_Cost_pc_of_properte_20`k')
1492  gen Time_Score`k' = (Max_Time`k'-Rg_Prp_Time_days_20`k')/(Max_Time`k'-Min_Time`k')
1493  gen Cost_Score`k' = (Max_Cost`k'-Rg_Prp_Cost_pc_of_properte_20`k')/(Max_Cost`k'-Min_Cost`k')
1494  gen Rg_Prp_score_`k' = (Time_Score`k' + Cost_Score`k')/2
1495  }
1496
1497  **Merging
1498  *****
1499  merge 1:1 Country using "${created_data}\IFAD_Score_Normalized.dta"
1500  gen LRA_score = (Rg_Prp_score_20 + IFAD_Score2020)/2
1501
1502  WBNaming Country
1503  merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1504
1505  save "${created_data}\LRA Full.dta", replace
1506  preserve
1507  keep if Income == "LISP"
1508  PercentRankInc LRA_score
1509  save "${created_data}\LRA LISP.dta", replace
1510  restore
1511  keep if Income == "HISP"
1512  PercentRankInc LRA_score
1513  save "${created_data}\LRA HISP.dta", replace
1514  append using "${created_data}\LRA LISP.dta"
1515
1516  keep Country Income percentile Score_In_FY20
1517  gen Indicator = "Land_Rights_And_Access"
1518  rename percentile Percentile
1519  save "${created_data}\Land_Rights_And_Access.dta", replace
1520  export excel using "${final_data}\FY20_Land_Rights_And_Access.xlsx", firstrow(variables)
1521  replace

```

```

1522
1523 ////////////////////////////////////////////////////
1524 //   WGI Control of Corruption   //
1525 ////////////////////////////////////////////////////
1526 //The following four indicators are drawn from the World Governance Indicators.
1527 //The code for each follows the same procedures
1528 /*
1529 //Download instructions:
1530 Go to https://info.worldbank.org/governance/wgi/ and download the excel file
1531 Name the file wgidataset.xlsx and save in the raw data folder
1532 */
1533
1534 import excel "${raw_data}\wgidataset.xlsx", sheet("ControlofCorruption") clear
1535 forvalues k = 1(1)13 { //Dropping extraneous header information
1536 drop in 1
1537 }
1538 foreach var of varlist _all {
1539 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1540 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1541 replace `var' = "Country" if `var' == "Country/Territory"
1542 }
1543 drop in 2 //Dropping type now that it is incorporated into the first row
1544 renvars, map (word(@[1],1))
1545 drop in 1
1546 destring, replace
1547 save "${created_data}\Control of Corruption Full.dta" , replace
1548
1549 **Merging
1550 *****
1551
1552 keep Country Estimate2018
1553 rename Estimate2018 Corruption_Score
1554
1555 WBNaming Country
1556 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1557 save "${created_data}\Corruption Full.dta", replace
1558
1559 //Calculating Percentile Ranks
1560 preserve
1561 keep if Income == "LISP"
1562 //For all the WGI indicators, we shift all values so that 0 is the median
1563 egen median = median(Corruption_Score)
1564 replace Corruption_Score = Corruption_Score - median
1565 drop median
1566 PercentRankInc Corruption_Score
1567 save "${created_data}\Corruption LISP.dta", replace
1568 restore
1569 keep if Income == "HISP"
1570 //For all the WGI indicators, we shift all values so that 0 is the median
1571 egen median = median(Corruption_Score)
1572 replace Corruption_Score = Corruption_Score - median
1573 drop median
1574 PercentRankInc Corruption_Score
1575 save "${created_data}\Corruption HISP.dta", replace
1576 append using "${created_data}\Corruption LISP.dta"
1577
1578 keep Country Income percentile Score_In_FY20
1579 gen Indicator = "Control_Of_Corruption"
1580 rename percentile Percentile
1581 save "${created_data}\Control Of Corruption.dta", replace
1582 export excel using "${final_data}\FY20 Control Of Corruption.xlsx", firstrow(variables)
replace
1583
1584
1585 ////////////////////////////////////////////////////
1586 //   WGI Government Effectiveness //
1587 ////////////////////////////////////////////////////
1588
1589 import excel "${raw_data}\wgidataset.xlsx", sheet("GovernmentEffectiveness") clear

```

```

1590 forvalues k = 1(1)13 { //Dropping extraneous header information
1591 drop in 1
1592 }
1593 foreach var of varlist _all {
1594 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1595 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1596 replace `var' = "Country" if `var' == "Country/Territory"
1597 }
1598 drop in 2 //Dropping type now that it is incorporated into the first row
1599 renvars, map (word(@[1],1))
1600 drop in 1
1601 destring, replace
1602 save "${created_data}\Government Effectiveness Full.dta" , replace
1603
1604 **Merging
1605 *****
1606
1607 keep Country Estimate2018
1608 rename Estimate2018 Gov_Eff_Score
1609
1610 WBNaming Country
1611 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1612
1613 //Calculating Percentiles
1614 preserve
1615 keep if Income == "LISP"
1616 //For all the WGI indicators, we shift all values so that 0 is the median
1617 egen median = median(Gov_Eff_Score)
1618 replace Gov_Eff_Score = Gov_Eff_Score - median
1619 drop median
1620 PercentRankInc Gov_Eff_Score
1621 save "${created_data}\Gov_Eff LISP.dta", replace
1622 restore
1623 keep if Income == "HISP"
1624 //For all the WGI indicators, we shift all values so that 0 is the median
1625 egen median = median(Gov_Eff_Score)
1626 replace Gov_Eff_Score = Gov_Eff_Score - median
1627 drop median
1628 PercentRankInc Gov_Eff_Score
1629 save "${created_data}\Gov_Eff HISP.dta", replace
1630 append using "${created_data}\Gov_Eff LISP.dta"
1631
1632 keep Country Income percentile Score_In_FY20
1633 gen Indicator = "Government_Effectiveness"
1634 rename percentile Percentile
1635 save "${created_data}\Government Effectiveness.dta", replace
1636 export excel using "${final_data}\FY20_Government_Effectiveness.xlsx", firstrow(variables)
replace

1637
1638
1639 ////////////////////////////////////////////////////
1640 //          WGI Rule of Law          //
1641 ////////////////////////////////////////////////////
1642
1643 import excel "${raw_data}\wgidataset.xlsx", sheet("RuleofLaw") clear
1644 forvalues k = 1(1)13 { //Dropping extraneous header information
1645 drop in 1
1646 }
1647 foreach var of varlist _all {
1648 replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1649 replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1650 replace `var' = "Country" if `var' == "Country/Territory"
1651 }
1652 drop in 2 //Dropping type now that it is incorporated into the first row
1653 renvars, map (word(@[1],1))
1654 drop in 1
1655 destring, replace
1656 save "${created_data}\Rule of Law Full.dta" , replace

```

```

1657
1658   keep Country Estimate2018
1659   rename Estimate2018 RuleofLaw_Score
1660
1661   WBNaming Country
1662   merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1663
1664   //Calculate percentiles
1665   preserve
1666   keep if Income == "LISP"
1667   //For all the WGI indicators, we shift all values so that 0 is the median
1668   egen median = median(RuleofLaw_Score)
1669   replace RuleofLaw_Score = RuleofLaw_Score - median
1670   drop median
1671   PercentRankInc RuleofLaw_Score
1672   save "${created_data}\RuleofLaw 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(RuleofLaw_Score)
1677   replace RuleofLaw_Score = RuleofLaw_Score - median
1678   drop median
1679   PercentRankInc RuleofLaw_Score
1680   save "${created_data}\RuleofLaw HISP.dta", replace
1681   append using "${created_data}\RuleofLaw LISP.dta"
1682
1683   keep Country Income percentile Score_In_FY20
1684   gen Indicator = "Rule_Of_Law"
1685   rename percentile Percentile
1686   save "${created_data}\Rule Of Law.dta", replace
1687   export excel using "${final_data}\FY20_Rule_Of_Law.xlsx", firstrow(variables) replace
1688
1689   //////////////////////////////////////
1690   //   WGI Regulatory Quality   //
1691   //////////////////////////////////////
1692
1693   import excel "${raw_data}\wgidataset.xlsx", sheet("RegulatoryQuality") clear
1694   forvalues k = 1(1)13 { //Dropping extraneous header information
1695   drop in 1
1696   }
1697   foreach var of varlist _all {
1698   replace `var' = `var'[2] + `var'[1] in 1 //Combining the year and the type (Estimate,
Standard Deviation, etc.) into a single variable
1699   replace `var' = "." if `var' == "#N/A" //Setting #N/A as missing
1700   replace `var' = "Country" if `var' == "Country/Territory"
1701   }
1702   drop in 2 //Dropping type now that it is incorporated into the first row
1703   renvars, map (word(@[1],1))
1704   drop in 1
1705   destring, replace
1706   save "${created_data}\Regulatory Quality Full.dta" , replace
1707
1708   keep Country Estimate2018
1709   rename Estimate2018 RegQuality_Score
1710
1711   WBNaming Country
1712   merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1713
1714   //Calculating Percentiles
1715   save "${created_data}\RegQuality Full.dta", replace
1716   preserve
1717   keep if Income == "LISP"
1718   //For all the WGI indicators, we shift all values so that 0 is the median
1719   egen median = median(RegQuality_Score)
1720   replace RegQuality_Score = RegQuality_Score - median
1721   drop median
1722   PercentRankInc RegQuality_Score
1723   save "${created_data}\RegQuality LISP.dta", replace
1724   restore
1725   keep if Income == "HISP"

```

```

1726 //For all the WGI indicators, we shift all values so that 0 is the median
1727 egen median = median(RegQuality_Score)
1728 replace RegQuality_Score = RegQuality_Score - median
1729 drop median
1730 PercentRankInc RegQuality_Score
1731 drop median
1732 save "${created_data}\RegQuality HISP.dta", replace
1733 append using "${created_data}\RegQuality LISP.dta"
1734
1735 keep Country Income percentile Score_In_FY20
1736 gen Indicator = "Regulatory_Quality"
1737 rename percentile Percentile
1738 save "${created_data}\Regulatory_Quality.dta", replace
1739 export excel using "${final_data}\FY20_Regulatory_Quality.xlsx", firstrow(variables) replace
1740
1741
1742
1743
1744 ////////////////////////////////////////////////////////////////////
1745 // Freedom of Information //
1746 ////////////////////////////////////////////////////////////////////
1747 // The Freedom of the Press indicator is a composite of three sub indicators
1748 // The main sub indicator is Freedom of the Press, with two minor sub indicators focused on
1749 // Right to Information, and Key Internet controls
1750 //The following code imports, cleans and combines these files to create scores for each
1751 // country.
1752
1753 ////////////////////////////////////////////////////////////////////
1754 // Freedom of the Press //
1755 ////////////////////////////////////////////////////////////////////
1756 // In previous years, the Freedom of the Press Indicator was drawn from Freedom House's
1757 // Freedom of the Press Report
1758 // However, Freedom House stopped producing this indicator in 2017. This required MCC to
1759 // find another source for these data
1760 // This year, MCC began using the Reporters Without Borders (RSF) Press Freedom Index in
1761 // place of Freedom House.
1762 // However, RSF was missing data for several small island nations, therefore MCC uses
1763 // percentile matching as shown below
1764 // to impute RSF scores for these countries based on data from previous years of Freedom
1765 // House.
1766 //Download Instructions:
1767 /*
1768 Go to the RSP website. Go to the Index details (https://rsf.org/en/ranking table), and
1769 select download the dataset
1770 Save as "RSF dataset 2019.csv"
1771 For the historical files: the "index_format_upload 2017.csv" can be found at the bottom of
1772 the page here "https://rsf.org/en/2017-press-freedom-index-ever-darker-world-map"
1773 And the classment2016 file can be found at the bottom of this page:
1774 "https://rsf.org/en/2016-world-press-freedom-index-leaders-paranoid-about-journalists"
1775 Go to the Freedom House website. Go to the Freedom of the Press Report
1776 (https://freedomhouse.org/report-types/freedom-press), and select "Freedom of the Press Data"
1777 Save as FOTP1980-FOTP2017_Public-Data.xlsx
1778 */
1779
1780 // The Following Sections format the RSF files for the Press Freedom Index
1781 //RSF Data Formatting
1782 set more off
1783 //RSF 2019
1784 import delimited "${raw_data}\RSF Dataset 2019.csv", clear
1785 keep en_country score2019
1786 rename en_country Country
1787 rename score2019 RSFscore
1788
1789 replace RSFscore = substr(RSFscore,".",1)
1790 destring, replace
1791 WBNaming Country
1792 save "${created_data}\RSF_Scores_2019.dta", replace
1793
1794 //RSF 2018
1795 import delimited "${raw_data}\RSF Dataset 2019.csv", clear

```

```

1785 keep en_country score2018
1786 rename en_country Country
1787 rename score2018 RSFscore
1788
1789 replace RSFscore = substr(RSFscore,"",".",..)
1790 destring, replace
1791 WBNaming Country
1792 save "${created_data}\RSF_Scores_2018.dta", replace
1793
1794 //RSF 2017
1795 import delimited "${raw_data}\index_format_upload 2017.csv", delimiter(";") clear
1796 //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). This has been confirmed with RSF.
1797 keep en_country overallscore2016 //This is actually 2017 data
1798 rename en_country Country
1799 rename overallscore2016 RSFscore
1800
1801 replace RSFscore = substr(RSFscore,"",".",..)
1802 destring, replace
1803 WBNaming Country
1804 save "${created_data}\RSF_Scores_2017.dta", replace
1805
1806 //RSF 2016
1807 import delimited "${raw_data}\index_format_upload 2017.csv", delimiter(";") clear
1808 //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).
1809 keep en_country score2015 //This is actually 2016 data
1810 rename en_country Country
1811 rename score2015 RSFscore
1812
1813 replace RSFscore = substr(RSFscore,"",".",..)
1814 destring, replace
1815 WBNaming Country
1816 save "${created_data}\RSF_Scores_2016.dta", replace
1817
1818 //RSF 2015
1819 import delimited "${raw_data}\classement2016.csv", delimiter(";") clear
1820 keep en_country score2015
1821 rename en_country Country
1822 rename score2015 RSFscore
1823
1824 replace RSFscore = substr(RSFscore,"",".",..)
1825 destring, replace
1826 WBNaming Country
1827 save "${created_data}\RSF_Scores_2015.dta", replace
1828
1829
1830 //The following section formats the Freedom House data
1831 //FH Data Formatting
1832 import excel "${raw_data}\FOTP1980-FOTP2017_Public-Data.xlsx", clear sheet(Data)
1833 replace A = "Country" in 5
1834 //The following section fills in missing year values for observations which were merged in
the original document
1835 qui ds //List all variables
1836 local variable_tally : word count `r(varlist)' //Count the number of variables
1837 local tally_minus = `variable_tally' - 1 //Create a macro which is one less than the full
variable count
1838 forvalues k = 1(1) `tally_minus' {
1839 local left : word `k' of `r(varlist)' //Finding the "left" value for the year observation
1840 local right : word `='k'+1' of `r(varlist)' //Finding the "right" value for the year
observation
1841 replace `right' = `left' if `right'[3] == "" in 3 //Filling forward the year values when
the values are missing
1842 }
1843 //Dropping Statuses and Sub-Scores
1844 foreach var of varlist all {
1845 if !(`var'[5] == "Print" | `var'[5] == "Broadcast" | `var'[5] == "Status" | `var'[5] ==
"Total Score" | `var'[5] == "Country") {

```

```

1846     drop `var'
1847 }
1848 }
1849
1850 //Dropping extraneous headings
1851 drop in 4
1852 drop in 2
1853 drop in 1
1854 replace A = "Country" in 1
1855 //The following section of code combines the year headings with the data type headings
1856 foreach var of varlist _all {
1857 label var `var' "`='var'[1]'"
1858 replace `var' = "Y`='var'[1]`_'='var'[2]'" in 1
1859 replace `var' = substr("`='var'[1]'", "-", "_", .) in 1
1860 }
1861 replace A = "Country" in 1
1862
1863 renvars, map (word(@[1],1))
1864 drop in 1
1865 drop in 1
1866
1867 destring, replace
1868 WBNaming Country
1869 save "${created_data}\Freedom_House_Press_Formatted.dta", replace
1870
1871
1872 **Imputation
1873 *****
1874 //Now that we have formatted the two files, we need to merge them and
1875 //use the freedom house data to impute the new scores using percentile
1876 //matching
1877
1878 set more off
1879 //This loop takes each RSF year and uses the freedom house data to impute the correct scores.
1880 forvalues RSF_year = 2015(1)2019 {
1881
1882 use "${created_data}\Freedom House Press Formatted.dta", clear
1883
1884 merge 1:1 Country using "${created_data}\RSF_Scores_`RSF_year'.dta", nogen keep(1 3)
1885 //Since the most recent Freedom House data is from 2017, when the RSF year is 2017, 2018,
1886 // or 2019,
1887 // we use that Freedom House data to impute the scores. However in earlier years,
1888 // we use matching year data:
1889 if "`RSF_year'" == "2019" | "`RSF_year'" == "2018" | "`RSF_year'" == "2017" {
1890 rename Y2017_Total FHScore
1891 }
1892 else {
1893 rename Y`RSF_year'_Total FHScore
1894 }
1895
1896 replace FHScore = "." if FHScore == "-"
1897 keep Country RSFscore FHScore
1898 destring, replace
1899 //In this line we are making the decision to only consider countries that are in the GNI
1900 //dataset as described above.
1901 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
1902 //In order to calculate the percentile matching scores, we need to calculate the number
1903 //of Freedom House Scores in the full dataset.
1904 egen n = count(FHScore)
1905 //We also need to calculate a ranking (1+ the number of scores that are lower)
1906 egen i = rank(FHScore), track
1907 //The Percentile rank in the full Freedom House dataset is then calculated as the rank
1908 //minus 1, divided by the count minus 1
1909 gen pcrank_FH = (i - 1) / (n - 1)
1910 drop i n
1911 egen RSF_count = count(RSFscore)
1912 //Now, we use the following calculations to determine where the rank of the freedom house
1913 //percentile
1914 //in the RSF dataset, and split it into its whole number and decimal part

```

```

1912 gen FH_Rank_in_RSF = (pcrank_FH*(RSF_count-1))+1
1913 gen FH_floor = floor(FH_Rank_in_RSF)
1914 gen FH_decimal = FH_Rank_in_RSF - FH_floor
1915
1916 sort RSFscore
1917 local RSF_cnt = `=RSF_count[1]`
1918 //This code creates a set of global macros corresponding to the ranking of the RSF scores
1919 qui forvalues k=1(1)`RSF_cnt' {
1920 local k_plus = `k' + 1
1921 global RSF_function_`k' = `=RSFscore[`k']`
1922 noisily di "`RSF_function_`k'" "`k'"
1923 }
1924 gen function_x = .
1925 gen function_x_plus = .
1926 local count = _N
1927 //The following code finds the RSF scores which rank on either side of
1928 //The Freedom House ranking, and then adding the product of the decimal
1929 //part of the Freedom House ranking and the difference of the two RSF scores
1930 //on either side.
1931 qui forvalues k = 1(1)`count' {
1932 replace function_x = ${RSF_function_`=FH_floor[`k']`} in `k'
1933 local floor_plus = `=FH_floor[`k']` + 1
1934 local count_minus = `count' -1
1935 if "`=FH_floor[`k']'" == "`RSF_cnt'" {
1936 noi di "`k'"
1937 local floor_plus = `=FH_floor[`k']`
1938 noisily di "`RSF_function_`floor_plus'"
1939 }
1940 replace function_x_plus = ${RSF_function_`floor_plus`} in `k'
1941 }
1942 gen New_RSF_Score = function_x + (FH_decimal*(function_x_plus-function_x))
1943 //This replaces the missing RSF scores with the newly imputed scores based on the FH data
1944 replace RSFscore = New_RSF_Score if missing(RSFscore)
1945 drop New_RSF_Score function_x_plus function_x FH_decimal FH_floor FH_Rank_in_RSF RSF_count
1946 pcrank_FH GNI FHScore
1947 rename RSFscore Y`RSF_year'_Total
1948 label var Y`RSF_year'_Total "`RSF_year'"
1949 gen Y`RSF_year'_Status = "NF"
1950 label var Y`RSF_year'_Status "`RSF_year'"
1951 WBNaming Country
1952 save "${created_data}\Freedom_of_The_Press_Imputed_`RSF_year'.dta", replace
1953 }
1954 //Merging each of the year files together
1955 use "${created_data}\Freedom_of_The_Press_Imputed_2015.dta", clear
1956 forvalues RSF_year = 2016(1)2019 {
1957 merge 1:1 Country using "${created_data}\Freedom_of_The_Press_Imputed_`RSF_year'.dta", nogen
1958 }
1959 save "${created_data}\FOP FY20.dta", replace
1960
1961 //Before we can create the final file, we need to process the Key Internet Controls and the
1962 //Right to Information files.
1963
1964 //Key Internet Controls
1965 //Now that the main indicator has been processed, we must process the other two minor
1966 //indicators
1967 //Countries gain 1 point (higher score indicates worse performance) for each key internet
1968 //control
1969 //on Freedom House's Key Internet controls metric up to 9.
1970 /*
1971 Download Instructions:
1972 Go to the report here
1973 https://www.freedomonthenet.org/sites/default/files/2019-11/11042019\_Report\_FH\_FOTN\_2019\_fina\_1\_Public\_Download.pdf
1974 On Page 28, there is a table with a list of countries and the number of key internet
1975 controls that they employed over the last year
1976 Copy the first two columns of this table into an excel file, label one column Country and
1977 the other KIC
1978 Label the sheet KIC table. Save the document as Key Internet Controls

```



```

1974
1975 */
1976 //
1977 import excel "${raw_data}\Key Internet Controls.xlsx", sheet("KIC table") firstrow clear
1978 WBNaming Country
1979 label var Country "Country"
1980 //Not all countries have KIC scores
1981 drop if Country == "" & KIC == .
1982 save "${created_data}\Key_Internet_Controls.dta", replace
1983
1984 //Before we can create the indicator, we need to process the Right to Information file.
1985
1986 ///////////////////////////////////////////////////
1987 // Right to Information //
1988 ///////////////////////////////////////////////////
1989 //Download Instructions: Go to the Country data page:
1990 https://www.rti-rating.org/country-data/
1991 // Click on "Excel" download. Save as RTI Ratings in the raw data folder
1992 import excel "${raw_data}\RTI Ratings.xlsx", sheet("Sheet1") firstrow clear
1993 keep Country Total
1994 WBNaming Country
1995 save "${created_data}\Right_To_Information.dta", replace
1996
1997 **Freedom of Information Merging
1998 *****
1999 //Merging files together:
2000 use "${created_data}\Right_To_Information.dta", clear
2001 merge 1:1 Country using "${created_data}\Key_Internet_Controls.dta", nogen
2002 merge 1:1 Country using "${created_data}\Freedom_of_The_Press_Imputed_2019.dta", nogen
2003 keep Country Total KIC Y2019_Total
2004 gen RTI_score = 4 if !missing(Total)
2005 replace KIC = 9 if !missing(KIC) & KIC>9
2006 recode RTI_score KIC (.=0)
2007 gen FOI_score = (Y2019_Total - RTI_score) + KIC
2008
2009 WBNaming Country
2010 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
2011 save "${created_data}\FOI Full.dta", replace
2012 preserve
2013 keep if Income == "LISP"
2014 PercentRankInc FOI_score
2015 replace percentile = 1-percentile //Note this is different for FOI due to the inverted scale
2016 save "${created_data}\FOI LISP.dta", replace
2017 restore
2018 keep if Income == "HISP"
2019 PercentRankInc FOI_score
2020 replace percentile = 1-percentile //Note this is different for FOI due to the inverted scale
2021 save "${created_data}\FOI HISP.dta", replace
2022 append using "${created_data}\FOI LISP.dta"
2023
2024 keep Country Income percentile Score_In_FY20
2025 gen Indicator = "Freedom_Of_Information"
2026 rename percentile Percentile
2027 save "${created_data}\Freedom Of Information.dta", replace
2028 export excel using "${final_data}\FY20_Freedom_Of_Information.xlsx", firstrow(variables)
2029 replace
2030
2031 ///////////////////////////////////////////////////
2032 // Trade Policy //
2033 ///////////////////////////////////////////////////
2034 /* Download Instructions:
2035 As of the publication of this do file, Heritage's scores for trade freedom are not public
2036 for all MCC candidate countries
2037 However, when they become public they will be available here
2038 https://www.heritage.org/index/download
2039 Select "Download Raw Data" and save the file as index2020 data.xls in the raw data folder
2040 */
2041
2042

```

```

2040 set more off
2041 import excel "${raw_data}\index2020_data.xls", clear firstrow
2042 keep CountryName TradeFreedom
2043 replace TradeFreedom = "." if TradeFreedom == "N/A"
2044 destring, replace
2045 keep CountryName TradeFreedom
2046 rename CountryName Country
2047 rename TradeFreedom Trade_Policy_Score
2048 WBNaming Country
2049
2050 **Merging
2051 *****
2052 WBNaming Country
2053 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
2054
2055 save "${created_data}\Trade Policy Full.dta", replace
2056 preserve
2057 keep if Income == "LISP"
2058 PercentRankInc Trade_Policy_Score
2059 save "${created_data}\Trade Policy LISP.dta", replace
2060 restore
2061 keep if Income == "HISP"
2062 PercentRankInc Trade_Policy_Score
2063 save "${created_data}\Trade Policy HISP.dta", replace
2064 append using "${created_data}\Trade Policy LISP.dta"
2065
2066 keep Country Income percentile Score_In_FY20
2067 gen Indicator = "Trade_Policy"
2068 rename percentile Percentile
2069 save "${created_data}\Trade Policy.dta", replace
2070 export excel using "${final_data}\FY20_Trade_Policy.xlsx", firstrow(variables) replace
2071
2072
2073 ////////////////////////////////////////////////////
2074 //      WEO/IMF Data      //
2075 ////////////////////////////////////////////////////
2076 /*
2077 Download instructions
2078 Use the October 2019 release
2079 Go to https://www.imf.org/external/pubs/ft/weo/2019/02/weodata/index.aspx
2080 Select by countries
2081 Select all countries
2082 Select Inflation, average consumer prices (Percent change) and General government net
lending/borrowing (Percent of GDP)
2083 Select Start 1980, end 2024
2084
2085 */
2086 set more off
2087 import excel "${raw_data}\WEO_Data.xlsx", clear
2088 //Cleaning up variables for renvars
2089 foreach var of varlist F-AX {
2090 replace `var' = "Y"=`var'[1]' in 1
2091 }
2092 foreach var of varlist _all {
2093 replace `var' = substr("`var'[1]'", " ", "_",.) in 1
2094 replace `var' = substr("`var'[1]'", "/", "_",.) in 1
2095 replace `var' = substr("`var'[1]'", "-", "_",.) in 1
2096
2097 replace `var' = "." if `var' == "n/a"
2098 replace `var' = "." if `var' == "---"
2099 }
2100 renvars, map (word(@[1],1))
2101 drop in 1
2102 destring, replace
2103 drop Y2019 Y2020 Y2021 Y2022 Y2023 Y2024 Estimates_Start_After
2104 foreach var of varlist Y* {
2105 lab var `var' `=substr("`var'", "Y", "",.)'
2106 }
2107 save "${created_data}\WEO_IMF_Data_Full.dta" , replace
2108

```

```

2109
2110 ////////////////////////////////////////////////////
2111 // Fiscal Policy //
2112 ////////////////////////////////////////////////////
2113 //This pulls directly from an IMF indicator
2114 set more off
2115 use "${created_data}\WEO_IMF_Data_Full.dta", clear
2116 keep if Subject_Descriptor == "General government net lending/borrowing"
2117 keep Country Y*
2118 WBNaming Country
2119
2120 **Merging
2121 *****
2122 //Fiscal Policy is a three year rolling average
2123 egen Fiscal_Policy_Score = rowmean(Y2016 Y2017 Y2018)
2124 keep Country Fiscal_Policy_Score
2125
2126 WBNaming Country
2127 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
2128
2129 save "${created_data}\Fiscal Policy Full.dta", replace
2130 preserve
2131 keep if Income == "LISP"
2132 PercentRankInc Fiscal_Policy_Score
2133 save "${created_data}\Fiscal Policy LISP.dta", replace
2134 restore
2135 keep if Income == "HISP"
2136 PercentRankInc Fiscal_Policy_Score
2137 save "${created_data}\Fiscal Policy HISP.dta", replace
2138 append using "${created_data}\Fiscal Policy LISP.dta"
2139
2140 keep Country Income percentile Score_In_FY20
2141 gen Indicator = "Fiscal_Policy"
2142 rename percentile Percentile
2143 save "${created_data}\Fiscal Policy.dta", replace
2144 export excel using "${final_data}\FY20_Fiscal_Policy.xlsx", firstrow(variables) replace
2145
2146 ////////////////////////////////////////////////////
2147 // Inflation //
2148 ////////////////////////////////////////////////////
2149 //This pulls directly from an IMF indicator
2150 set more off
2151 use "${created_data}\WEO IMF Data Full.dta", clear
2152 keep if Subject_Descriptor == "Inflation, average consumer prices"
2153 keep Country Y*
2154 WBNaming Country
2155
2156 **Merging
2157 *****
2158 rename Y2018 Inflation_Rate
2159 keep Country Inflation_Rate
2160
2161 WBNaming Country
2162 merge 1:1 Country using "${created_data}\GNI_FY20.dta", nogen keep(3)
2163
2164 save "${created_data}\Inflation Full.dta", replace
2165 preserve
2166 keep if Income == "LISP"
2167 PercentRankInc Inflation_Rate
2168 replace percentile = 1-percentile //Note this is different for Inflation due to the
inverted scale
2169 save "${created_data}\Inflation LISP.dta", replace
2170 restore
2171 keep if Income == "HISP"
2172 PercentRankInc Inflation_Rate
2173 replace percentile = 1-percentile //Note this is different for Inflation due to the
inverted scale
2174 save "${created_data}\Inflation HISP.dta", replace
2175 append using "${created_data}\Inflation LISP.dta"
2176

```

```
2177 keep Country Income percentile Score_In_FY20
2178 gen Indicator = "Inflation"
2179 rename percentile Percentile
2180 save "${created_data}\Inflation.dta", replace
2181 export excel using "${final_data}\FY20_Inflation.xlsx", firstrow(variables) replace
2182
2183
2184
2185
2186
```