

```

1: #Packages-----
2: library(ggplot2)
3: library(tidyverse)
4:
5: setwd("C:/Nishimoto/R/WBAL_R02/2_data/REF")
6: BaseYear <- 2010 # %>% as.numeric() # 基準年値
7: Year5 <- c(0, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015,
8:           2020, 2025, 2030, 2035, 2040, 2045, 2050, 2055, 2060, 2065,
9:           2070, 2075, 2080, 2085, 2090, 2095, 2100) # %>% as.character()
10:
11: while (0) {
12:   # 単位の連想配列>ファイル名にマッチさせる予定
13:   df_unit <- read.delim(file="./unit.txt", header=T)
14:   # view(df_unit)
15:   Unit_of_Var <- df_unit$unit
16:   names(Unit_of_Var) <- df_unit$filename
17:   # Unit_of_Var[names(Unit_of_Var)]
18: }
19:
20: # 国コードの連想配列
21: df_CC <- read.delim(file="./CC.txt", header=T)
22: df_CC <- rename(df_CC, 'Country'='IEA国名')
23: # view(df_CC)
24: Region_Code <- df_CC$AIM17
25: names(Region_Code) <- df_CC$Country
26: # Region_Code[names(Region_Code)]
27:
28: # タイトル行(ダミー)の作成
29: df_past <- read_csv("./POP_IEA.csv")
30: df_past <- df_past %>% mutate('REGION'='region', 'VARIABLE'='variable'
31:                               ) %>% rename('Country'='TIME') # 'TIME' OR 'X1'
32: df_past <- df_past[1,c(ncol(df_past),ncol(df_past)-1,1:(ncol(df_past)-2))] # 列の入替
33: # View(df_past)
34:
35: # 1ファイル毎に追加
36: files <- list.files() # 指定ディレクトリのファイル一覧を代入
37: for (file.name in files) {
38:   if ( regexpr('¥¥.csv$', file.name) < 0 ) { next }
39:   d <- read_csv(file.name) # ファイルを仮変数に読み込む
40:   file.name <- gsub(".csv", "", file.name)
41:   d <- d %>% rename(REGION=TIME
42:                     ) %>% mutate(VARIABLE=file.name)
43:
44:   # 国コード付与
45:   d <- d %>% mutate(AIM17=Region_Code[d$REGION]
46:                     ) %>% rename('Country'='REGION'
47:                                     ) %>% rename('REGION'='AIM17'
48:                                                     ) %>% drop_na('REGION') # 国コードのない行は無視
49:   d <- d %>% mutate(Country = str_replace_all(Country,
50:                                                 pattern = c("Memo.: ""="", ""Memo: ""="", "" .if no detail.""="")))
51:   # d <- d[1,c(ncol(d),1:(ncol(d)-1))] # 列の入替
52:   df_past <- rbind(df_past, d)
53: }
54: df_past <- df_past %>% filter(REGION!='region') # ダミー行のデータを削除
55:
56: # View(df_past)
57: write_csv(df_past, "./../df_past_written_everyYear.csv") # VARIABLE REGION Country
58:
59: Titlerow1 <- c('MODEL','SCENARIO','REGION','VARIABLE','UNIT')
60: Titlerow2 <- c('REGION','Country','VARIABLE') # 'SCENARIO'は別
61: Titlerow3 <- c('SCENARIO','Country')
62:
63: # (課題)df_past の基準年値を補間する>補間後に5年置きにする
64:
65: # while (0) { # df_past を5年置きにする
66: # 列名から5年置きの年を取得>先送り>直接入力(仮)
67: # df_past <- df_past %>% select(all_of(Titlerow2), all_of(Year5)) # 後工程で処理する
68: df_past <- df_past %>% mutate(SCENARIO='Historical') # 書式を揃える
69: # View(df_past)
70: # } # df_past を5年置きにする
71:
72: scenarioname <- 'Baseline' # 読込対象の将来シナリオ(今は読込の時点でシナリオを絞っている)
73:

```

```

74: # while (0) { # 将来シナリオの読込
75: df_future <- read_csv("C:/_Nishimoto/R/WBAL_R02/2_data/REF2/IAMCTemplate.csv")
76: # View(df_future)
77:
78: df_future <- df_future %>% select(-c('MODEL','UNIT'))
79: ) %>% filter(SCENARIO == scenarioname # rbind前にシナリオを絞る場合
80: ) %>% filter(!REGION %in% c('ASIA2','World'))
81: ) %>% mutate(Country = REGION) # 書式を揃える #シナリオ名
82:
83: # IAMCTemplate の名前を IEA に揃える@ 'I'対策
84: df_future <- df_future %>% mutate(VARIABLE = str_replace_all(VARIABLE, pattern = c(
85: 'GDP.MER' = 'GDP IEA',
86: 'Population' = 'POP IEA',
87: 'Primary Energy' = 'TES_Total',
88: 'Emissions.CO2.Energy' = 'CO2_fuel_Total',
89: 'Final Energy.Electricity' = 'TFC_Elec_Total',
90: 'Final Energy.Industry.Electricity' = 'TFC_Elec_Ind',
91: 'Final Energy.Transportation.Electricity' = 'TFC_Total_Tra',
92: 'Final Energy.Residential.Electricity' = 'TFC_Elec_Res',
93: 'Final Energy.Commercial.Electricity' = 'TFC_Elec_Com',
94: 'Final Energy' = 'TFC_Total_Total' )))
95: # View(df_future)
96: write_csv(df_future, "../df_future_written.csv")
97: # } # 将来シナリオの読込
98:
99: df_long <- rbind(gather(df_past, key="Year", value="Value", -all_of(Titlerow2), -SCENARIO),
100: gather(df_future, key="Year", value="Value", -all_of(Titlerow2), -SCENARIO))
101: df_long$Year <- as.numeric(df_long$Year)
102: df_long$Value <- as.numeric(df_long$Value) # NA warning > 確認済
103: write_csv(df_long, "../df_long_written.csv")
104:
105: # 指標の処理 # Variable_Names_for_Indicators df_vni <- indicator, numerator, denominator
106: df_vni <- matrix(c(
107: 'GDP_Capita', 'GDP IEA', 'POP IEA',
108: 'Energy_Intensity', 'TES_Total', 'GDP IEA',
109: 'Carbon_Intensity', 'CO2_fuel_Total', 'TES_Total',
110: 'Electricity_Rate_Total', 'TFC_Elec_Total', 'TFC_Total_Total',
111: 'Electricity_Rate_Ind', 'TFC_Elec_Ind', 'TFC_Total_Ind',
112: 'Electricity_Rate_Tra', 'TFC_Elec_Tra', 'TFC_Total_Tra',
113: 'Electricity_Rate_Res', 'TFC_Elec_Res', 'TFC_Total_Res',
114: 'Electricity_Rate_Com', 'TFC_Elec_Com', 'TFC_Total_Com'),
115: ncol=8, nrow=3)
116:
117: for (dummyloop in 1) { # 国名のみのダミー列の作成
118: df_Graph <- df_long %>% select(c('Country')) %>% arrange(Country) %>% distinct()
119: } # ダミー列の作成
120:
121:
122: for (i in 1:ncol(df_vni)) { # 指標毎の処理 # テスト後に戻す
123:
124: indicator <- df_vni[1,i]
125: numerator <- df_vni[2,i]
126: denominator <- df_vni[3,i]
127:
128: for (variable_name in c(numerator, denominator)) {
129: df_toMerge <- df_long %>% filter(VARIABLE==variable_name
130: ) %>% select(-c('VARIABLE'))
131: ) %>% arrange(Year)
132: df_toMerge <- eval(parse(text=paste0("rename(df_toMerge,", variable_name,"=Value)"))))
133: # View(df_toMerge)
134: df_Graph <- merge(df_Graph, df_toMerge)
135: }
136: df_Graph <- df_Graph %>% drop_na('REGION','Year') # ダミー列のデータを削除
137:
138: # 指標の算出
139: df_Graph <- eval(parse(text=paste0(
140: "df_Graph %>% mutate(",indicator,"=",numerator,"/",denominator,")"))))
141:
142: # 指標の基準年値 I(t=BaseYear) # 基準年データがない国の処理
143: Sample_Country <- c('Former Soviet Union','Former Yugoslavia','South Sudan','Bosnia and Herzegovina') #
GDP(2010)が無い国
144: Interpolate_NA <- 'fill_latest_or_first_existing_value'
145: for (dummyloop in 1) { # 基準年データがない国の処理

```

```

146: df_Graph_interpolated <- eval(parse(text=paste0(
147:   "df_Graph %>% group_by(Country
148:   ) %>% mutate(Indicator2=",indicator,"
149:   ) %>% fill(Indicator2, .direction='down' # 前年値を優先
150:   ) %>% fill(Indicator2, .direction='up' # 前年値がなければ後年値
151:   ) %>% mutate(SCENARIO2=if_else(is.na(",indicator,"), Interpolate_NA, SCENARIO)))"))
152:
153: df_Graph_BaseYear <- df_Graph_interpolated %>% filter(Year==BaseYear
154:   ) %>% mutate(Year=0
155:   ) %>% select(-Indicator2, -SCENARIO2)
156:
157: # 補完値の確認用 時系列XY散布図 by サンプル国
158: df_Graph_interpolated <- df_Graph_interpolated %>% filter(Country %in% Sample_Country)
159: g <- ggplot(df_Graph_interpolated, aes(x=Year,y=Indicator2,
160:   color=Country, shape=SCENARIO2)) +
161:   geom_point() +
162:   # theme(legend.position='none') +
163:   ylab(indicator) +
164:   scale_shape_manual(values=c(24,19))
165: plot(g)
166: filename <- paste("Interpolated_",indicator,"_",Interpolate_NA, sep="")
167: ggsave(file=paste("C:/Nishimoto/R/WBAL_R02/4_output/test/",filename,".png", sep=""))
168:
169: } # 基準年データがない国の処理
170:
171: for (dummyloop in 1) { # 基準年値をdf_Graphに追加する(0年値として追加)
172:
173:   df_Graph <- df_Graph %>% rbind(df_Graph_BaseYear
174:   ) %>% filter(Year %in% Year5
175:   ) %>% group_by(Country
176:   ) %>% arrange(Country, Year)
177:   df_Graph <- eval(parse(text=paste0(
178:     "df_Graph %>% mutate(",indicator,"_scaled=",indicator,"/",indicator,"[Year==0]
179:     ) %>% filter(Year!=0)"
180:   )))
180:   # indicator_scaled = I(t)/I(t=BaseYear)
181:   # df_Graph <- df_Graph %>% filter(Year!=0)
182:
183: } # 基準年値をdf_Graphに追加する
184:
185: for (dummyloop in 1) { # 指標の変化率
186:
187:   # 指標の変化率 ChangeRate_Indicator=(I(t)-I(t-1))/I(t=BaseYear)/((t)-(t-1))
188:   df_Graph <- eval(parse(text=paste0(
189:     "df_Graph %>% group_by(Country
190:     ) %>% arrange(Country, Year
191:     ) %>% mutate(ChangeRate_",indicator,"
192:     "=(",indicator,"_scaled-lag(",indicator,"_scaled, n=1))/(Year-lag(Year, n=1))
193:     ) %>% ungroup(
194:     )"
195:   )))
196:
197: } # 指標の変化率
198:
199: while (0) { # 指標の変化率bk
200:
201:   # 指標の変化率 ChangeRate_Indicator=(I(t)-I(t-1))/I(t=BaseYear)/((t)-(t-1))
202:   df_Graph <- eval(parse(text=paste0(
203:     "df_Graph %>% group_by(Country
204:     ) %>% arrange(Country, Year
205:     ) %>% mutate(Year_pre=lag(Year, n=1)
206:     ) %>% mutate(",indicator,"_pre=lag(",indicator,"_pre, n=1)
207:     ) %>% mutate(RatePre_",indicator,"=",indicator,"/",indicator,"_pre
208:     ) %>% mutate(ChangeRate_",indicator,"
209:     "=(",indicator,"-",indicator,"_pre)/(Year", "-", "Year", "_pre)",
210:     ")",indicator,"_pre",
211:     )"
212:   )))
213: } # 指標の変化率bk
214:
215: } # 指標毎の処理
216: # View(df_toMerge)
217: View(df_Graph)
218: write_csv(df_Graph, "../df_Graph_written.csv")

```

```

219: setwd("C:/Nishimoto/R/WBAL_R02/4_output/test/")
220: while (0) { # グラフ出力 for (dummyloop in 1)
221:
222:   # scenarionames <- c('Baseline','2C') # c('Baseline','2C','1.5C','2.5C','WB2C')
223:   scenarionames <- c('Baseline')
224:   # indicators <- c('GDP_Capita') # テスト中
225:
226:   indicators <- c('GDP_Capita',
227:                 'Energy_Intensity_scaled','ChangeRate_Energy_Intensity',
228:                 'Carbon_Intensity_scaled','ChangeRate_Carbon_Intensity',
229:                 'Electricity_Rate_Total_scaled','ChangeRate_Electricity_Rate_Total',
230:                 'Electricity_Rate_Ind_scaled','ChangeRate_Electricity_Rate_Ind')
231:
232:   # 出力対象のXY軸を指定する x_names(n) vs y_names(n)のグラフが出力される
233:
234:   x_names <- c(rep('Year',length(indicators)),
235:               rep('GDP_Capita',length(indicators)-1),
236:               rep('REGION',length(indicators)-1)
237: )
238:   y_names <- c(indicators,
239:               indicators[-1],
240:               indicators[-1])
241:   y2_names <- indicators[c(3,5,7,9)]
242:
243:   for (scenarioname in scenarionames) {
244:
245:     # XY散布図 by 17地域
246:     pdf(file=paste("../",scenarioname,"_XY.pdf", sep=""))
247:     for (num in 1:length(x_names)) {
248:
249:       g <- eval(parse(text=paste0(
250:         "ggplot(df_Graph, aes(x=",x_names[num],"",y=",y_names[num],
251:         ",color=REGION, shape=SCENARIO)) +
252:       #   geom_line() +
253:       geom_point() +
254:       scale_shape_manual(values=c(19,21))"))
255:       plot(g)
256:       filename <- paste(scenarioname,num,"_",x_names[num],"-",y_names[num], sep="")
257:       ggsave(file=paste("../png/",filename,".png"))
258:     }
259:     dev.off()
260:
261:     # XY散布図 by 国別
262:     pdf(file=paste("../",scenarioname,"_XY_Country.pdf", sep=""))
263:     for (num in 1:length(x_names)) {
264:
265:       g <- eval(parse(text=paste0(
266:         "ggplot(df_Graph, aes(x=",x_names[num],"",y=",y_names[num],
267:         ",color=Country, shape=SCENARIO)) +
268:       geom_line() +
269:       geom_point() +
270:       theme(legend.position='none') +
271:       scale_shape_manual(values=c(19,21))"))
272:       plot(g)
273:       filename <- paste(scenarioname,"_",num,"_",x_names[num],"-",y_names[num],"_CN", sep="")
274:       ggsave(file=paste("../png/",filename,".png"))
275:     }
276:     dev.off()
277:
278:     # 箱ヒゲ図
279:     pdf(file=paste("../",scenarioname,"_boxplot.pdf", sep=""))
280:     for (indicator in indicators) {
281:
282:       g <- eval(parse(text=paste0(
283:         "ggplot(df_Graph, aes(x=", "REGION", "",y=",indicator,
284:         ",color=SCENARIO)) +
285:       geom_boxplot() +
286:       geom_jitter(shape=20, position=position_dodge(0.8))"))
287:       plot(g)
288:       filename <- paste(scenarioname,"_", "boxplot_",indicator, sep="")
289:       ggsave(file=paste("../png/",filename,".png", sep=""), width=6, height=4, dpi=100)
290:     }
291:     dev.off()

```

```

292:
293: # 頻度分布
294: pdf(file=paste("./",scenarioname,"_histogram.pdf", sep=""))
295: for (indicator in indicators) {
296:
297:   g <- eval(parse(text=paste0(
298:     "ggplot(df_Graph, aes(x=",indicator,
299:     ",color=SCENARIO)) +
300:     geom_histogram(bins=50) +
301:     ylab('Count of Region-Year')"))))
302:   plot(g)
303:   filename <- paste(scenarioname,"_", "histogram_",indicator, sep="")
304:   ggsave(file=paste("./png/",filename,".png", sep=""), width=6, height=4, dpi=100)
305: }
306: dev.off()
307:
308: # 確率密度分布
309: pdf(file=paste("./",scenarioname,"_density.pdf", sep=""))
310: for (indicator in indicators) {
311:
312:   g <- eval(parse(text=paste0(
313:     "ggplot(df_Graph, aes(x=",indicator,
314:     ",color=SCENARIO)) +
315:     geom_density() +
316:     ylab('Density (Count scaled to 1) of Region-Year')"))))
317:   plot(g)
318:   filename <- paste(scenarioname,"_", "density_",indicator, sep="")
319:   ggsave(file=paste("./png/",filename,".png", sep=""), width=6, height=4, dpi=100)
320: }
321: dev.off()
322:
323: }
324: } # グラフ出力

```