

```

1: # Packages -----
2: library(ggplot2)
3: library(ggthemes)
4: library(tidyverse)
5:
6: root <- 'C:/_Nishimoto/R/WBAL_R02/'
7: Titlerow1 <- c('MODEL','SCENARIO','REGION','VARIABLE','UNIT')
8: Titlerow2 <- c('REGION','Country','VARIABLE','SCENARIO')
9: Titlerow3 <- c('SCENARIO','Country','REGION','Year')
10: 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)
11: BaseYear <- 2010 # %>% as.numeric() # 基準年値
12: Sample_Country <- c('Former Soviet Union','Former Yugoslavia','South Sudan','Bosnia and Herzegovina') #
  GDP(2010)が無い国
13: Interpolate_NA <- 'fill' # 'fill latest or first existing value'
14: Year5 <- c(1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015,
15:           2020, 2025, 2030, 2035, 2040, 2045, 2050, 2055, 2060, 2065,
16:           2070, 2075, 2080, 2085, 2090, 2095, 2100) # %>% as.character()
17: # Year==0 as Base-Year
18:
19: # Past -----
20: setwd(paste(root,"2_data/REF", sep=""))
21:
22: while (0) {
23:   # 単位の連想配列>ファイル名にマッチさせる予定
24:   df_unit <- read.delim(file="./unit.txt", header=T)
25:   # view(df_unit)
26:   Unit_of_Var <- df_unit$unit
27:   names(Unit_of_Var) <- df_unit$filename
28:   # Unit_of_Var[names(Unit_of_Var)]
29: }
30:
31: # 国コードの連想配列
32: df_CC <- read.delim(file="./CC.txt", header=T)
33: df_CC <- rename(df_CC, 'Country'='IEA国名')
34: # view(df_CC)
35: Region_Code <- df_CC$AIM17
36: names(Region_Code) <- df_CC$Country
37: # Region_Code[names(Region_Code)]
38:
39: # タイトル行(ダミー)の作成
40: df_past <- read_csv("./POP_IEA.csv")
41: df_past <- df_past %>% mutate('REGION'='region', 'VARIABLE'='variable'
42:                               ) %>% rename('Country'='TIME') # 'TIME' OR 'X1'
43: df_past <- df_past[1,c(ncol(df_past),ncol(df_past)-1,1:(ncol(df_past)-2))] # 列の入替
44: # View(df_past)
45:
46: # 1ファイル毎に追加
47: files <- list.files() # 指定ディレクトリのファイル一覧を代入
48: for (file_name in files) {
49:   if (regexpr('¥¥.csv$', file_name) < 0 ) { next }
50:   d <- read_csv(file_name) # ファイルを仮変数に読み込む
51:   file_name <- gsub("¥¥.csv", "", file_name)
52:   d <- d %>% rename(REGION=TIME
53:                     ) %>% mutate(VARIABLE=file_name)
54:
55:   # 国コード付与
56:   d <- d %>% mutate(AIM17=Region_Code[d$REGION]
57:                     ) %>% rename('Country'='REGION'
58:                     ) %>% rename('REGION'='AIM17'
59:                     ) %>% drop_na('REGION') # 国コードのない行は無視
60:   # d <- d[1,c(ncol(d),1:(ncol(d)-1))] # 列の入替
61:   df_past <- rbind(df_past, d)
62: }
63: setwd(paste(root,"4_output/", sep=""))
64:
65: df_past <- df_past %>% filter(REGION!='region') # ダミー行のデータを削除

```

```

66: df_past <- df_past %>% mutate(SCENARIO='Historical') # 書式を揃える
67: df_past <- df_past %>% mutate(Country = str_replace_all(Country,
68:   pattern = c("Memo: " = "", "Memo: " = "", ".if no detail." = "")))
69: write_csv(df_past, "../df_past_written_everyYear.csv") # VARIABLE REGION Country
70:
71: df_past_long <- df_past %>% gather(key='Year', value='Value', -all_of(Titlerow2))
72: df_past_long$Year <- as.numeric(df_past_long$Year)
73: df_past_long$Value <- as.numeric(df_past_long$Value) # NA warning > 確認済
74:
75: # Fill past -----
76: # 基準年データがない国の処理 for (dummyloop in 1) {
77:
78: df_past_long <- df_past_long %>% group_by(VARIABLE, Country
79:   ) %>% arrange(VARIABLE, Country, Year
80:   ) %>% mutate(Value2=Value
81:   ) %>% fill(Value2, .direction='down' # 前年値を優先
82:   ) %>% fill(Value2, .direction='up' # 前年値がなければ後年値
83:   ) %>% mutate(SCENARIO2=if_else(is.na(Value), Interpolate_NA, SCENARIO)
84:   ) %>% ungroup()
85:
86: df_past_BaseYear <- df_past_long %>% filter(Year==BaseYear
87:   ) %>% mutate(Year=0)
88:
89: while (0) { # PDF出力
90:   pdf(file=paste("../", "past_filled.pdf", sep="")) # PDF出力開始
91:   for (y_name in unique(df_past_long$VARIABLE)) { # 補完値の確認出力 by サンプル国
92:     df_Graph_past <- df_past_long %>% filter(Country %in% Sample_Country
93:       ) %>% filter(VARIABLE==y_name)
94:     g <- ggplot(df_Graph_past, aes(x=Year,y=Value2,
95:       color=Country, shape=SCENARIO2)) +
96:       geom_point() +
97:       # theme(legend.position='none') +
98:       ylab(y_name) +
99:       scale_shape_manual(values=c(24,19))
100:    plot(g)
101:    filename <- paste("Interpolated_", y_name, "_", Interpolate_NA, sep="")
102:    # ggsave(file=paste("../png/", filename, ".png", sep=""))
103:  }
104:  dev.off() # PDF出力終了
105: } # PDF出力
106:
107: # 基準年データがない国の処理 }
108:
109: write_csv(df_past_long, "../df_past_long_written_everyYear.csv") # VARIABLE REGION Country
110: write_csv(df_past, "../df_past_written_everyYear.csv") # VARIABLE REGION Country
111:
112: # Future -----
113: # while (0) { # 将来シナリオの読込
114: df_future <- read_csv(paste(root, "2_data/REF2/", "IAMCTemplate.csv", sep=""))
115: # View(df_future)
116:
117: df_future <- df_future %>% select(-c('MODEL','UNIT'))
118: # ) %>% filter(SCENARIO == scenarioname # rbind前にシナリオを絞る場合
119: ) %>% filter(IREGION %in% c('ASIA2', 'World'))
120: ) %>% mutate(Country = REGION) # 書式を揃える # 国名=地域名
121:
122: df_future <- df_future %>% mutate('0'=as.numeric(df_future$'2010')) # 基準年値をコピーしておく
123:
124: # IAMCTemplate(future) の名前を IEA(past) に揃える@ 'I'対策 # recode でも出来るらしい
125: df_future <- df_future %>% mutate(VARIABLE = str_replace_all(VARIABLE, pattern = c(
126:   'GDP.MER' = 'GDP_IEA',
127:   'Population' = 'POP_IEA',
128:   'Primary Energy' = 'TES_Total',
129:   'Emissions.CO2.Energy' = 'CO2_fuel_Total',
130:   'Final Energy.Industry.Electricity' = 'TFC_Elec_Ind',
131:   'Final Energy.Transportation.Electricity' = 'TFC_Elec_Tra',

```

```

132: 'Final Energy.Residential.Electricity' = 'TFC_Elec_Res',
133: 'Final Energy.Commercial.Electricity' = 'TFC_Elec_Com',
134: 'Final Energy.Electricity' = 'TFC_Elec_Total',
135: 'Final Energy.Industry' = 'TFC_Total_Ind',
136: 'Final Energy.Transportation' = 'TFC_Total_Tra',
137: 'Final Energy.Residential' = 'TFC_Total_Res',
138: 'Final Energy.Commercial' = 'TFC_Total_Com',
139: 'Final Energy' = 'TFC_Total_Total')) # 順番注意／.最長マッチ
140: # View(df_future)
141: write_csv(df_future, "../df_future_pre_written.csv")
142: # ここで、上記以外のデータは捨てる
143: df_future <- df_future %>% filter(VARIABLE %in%
144:   c('GDP IEA', 'POP IEA', 'TES_Total', 'CO2_fuel_Total',
145:     'TFC_Total_Total', 'TFC_Total_Ind', 'TFC_Total_Tra', 'TFC_Total_Res', 'TFC_Total_Com',
146:     'TFC_Elec_Total', 'TFC_Elec_Ind', 'TFC_Elec_Tra', 'TFC_Elec_Res', 'TFC_Elec_Com'))
147: write_csv(df_future, "../df_future_written.csv")
148: # } # 将来シナリオの読込
149:
150: # Connect Past & Future -----
151: df_long_past <- df_past_long %>% rbind(df_past_BaseYear)
152: df_long_past <- df_long_past %>% select(-Value2, -SCENARIO2
153:   ) %>% filter(Year %in% c(Year5, 0)) # 各年から5年置きに
154: df_long_future <- gather(df_future, key=Year, value=Value, -all_of(TitleRow2))
155: df_long_future$Year <- as.numeric(df_long_future$Year)
156: df_long <- rbind(df_long_past, df_long_future)
157: df_long$Year <- as.numeric(df_long$Year)
158: df_long$Value <- as.numeric(df_long$Value) # NA warning > 確認済
159: write_csv(df_long, "../df_long_written.csv")
160:
161: # Aggregation to Region -----
162: for (dummyloop in 1) { # 地域集約 # while (0)
163:   df_long_agg <- aggregate(Value~VARIABLE+REGION+SCENARIO+Year, df_long, sum) # 集約対象=Country
164:   df_long_agg <- df_long_agg %>% mutate(Country=REGION)
165:   # df_long <- df_long_agg # 過去17地域、将来17地域で出力する場合
166:   for (dummyloop in 1) { # 過去17地域と過去約2百数十ヶ国を併記する場合
167:     df_long_agg <- df_long_agg %>% filter(SCENARIO=='Historical'
168:       ) %>% mutate(SCENARIO='Historical_R17')
169:     df_long <- df_long %>% rbind(df_long_agg)
170:   } # 併記する場合
171:   write_csv(df_long_agg, "../df_long_agg_written.csv")
172: } # 地域集約
173:
174:
175: # Table format and Indicator -----
176: # 指標の処理 # Variable_Names_for_Indicators df_vni <- indicator, numerator, denominator
177: df_vni <- matrix(c(
178:   'GDP_Capita', 'GDP IEA', 'POP IEA',
179:   'Energy_Intensity', 'TES_Total', 'GDP IEA',
180:   'Carbon_Intensity', 'CO2_fuel_Total', 'TES_Total',
181:   'Electricity_Rate_Total', 'TFC_Elec_Total', 'TFC_Total_Total',
182:   'Electricity_Rate_Ind', 'TFC_Elec_Ind', 'TFC_Total_Ind',
183:   'Electricity_Rate_Tra', 'TFC_Elec_Tra', 'TFC_Total_Tra',
184:   'Electricity_Rate_Res', 'TFC_Elec_Res', 'TFC_Total_Res',
185:   'Electricity_Rate_Com', 'TFC_Elec_Com', 'TFC_Total_Com'),
186:   ncol=8, nrow=3)
187:
188: for (dummyloop in 1) { # 国名のみのダミー列の作成
189:   df_Graph <- df_long %>% select(c('Country')) %>% arrange(Country) %>% distinct()
190: } # ダミー列の作成
191:
192: for (i in 1:ncol(df_vni)) { # 指標毎の処理1 # テスト後に戻す (i in 1:ncol(df_vni))
193:   indicator <- df_vni[1,i]
194:   numerator <- df_vni[2,i]
195:   denominator <- df_vni[3,i]
196:
197:   for (variable_name in c(numerator, denominator)) {

```

```

198: df_toJoin <- df_long %>% filter(VARIABLE==variable_name
199: ) %>% select(-c('VARIABLE'))
200: ) %>% arrange(Year)
201: df_toJoin <- eval(parse(text=paste0("df_toJoin %>% rename(",variable_name,"=Value)")))
202: # View(df_toJoin)
203: df_Graph <- df_Graph %>% full_join(df_toJoin)
204: }
205: df_Graph <- df_Graph %>% drop_na('REGION','Year') # ダミー列のデータを削除
206: df_Graph <- eval(parse(text=paste0(
207: "df_Graph %>% mutate(",indicator,"=",numerator,"/",denominator,")"))) # 指標の算出
208:
209: for (dummyloop in 1) { # 基準年値をdf_Graphに追加する(0年値として追加)
210:
211:   df_Graph <- df_Graph %>% group_by(SCENARIO,Country) %>% arrange(SCENARIO,Country,Year)
212:   df_Graph <- eval(parse(text=paste0(
213: "df_Graph %>% mutate(",indicator,"_scaled=",indicator,"/",indicator,"[Year==0])"
214: ))) # indicator_scaled = I(t)/I(t=BaseYear)
215:   df_Graph <- df_Graph %>% ungroup()
216:
217: } # 基準年値をdf_Graphに追加する
218:
219: } # 指標毎の処理1
220: df_Graph$SCENARIO <- factor(df_Graph$SCENARIO,
221:   levels=c('Historical','Historical R17','Baseline','2.5C','2C','1.5C','WB2C'))
222: df_Graph <- df_Graph %>% filter(Year!=0) %>% group_by(SCENARIO,Country) %>%
  arrange(SCENARIO,Country,Year)
223: write_csv(df_Graph, "../df_Graph_afterfulljoin_written.csv")
224:
225: # Change rate -----
226: for (i in 1:ncol(df_vni)) { # 指標毎の処理2 # テスト後に戻す (i in 1:ncol(df_vni))
227:
228:   indicator <- df_vni[1,i]
229:
230:   # 指標の変化率(t年比) ChangeRate_Indicator=(I(t)-I(t-1))/({I(t)+I(t-1)}/2)/((t)-(t-1))
231:   df_Graph <- eval(parse(text=paste0(
232: "df_Graph %>% mutate(ChangeRateBY_",indicator,
233: "=(",indicator,"_scaled-lag(",indicator,"_scaled, n=1))/(Year-lag(Year, n=1))
234: ) %>% mutate(ChangeRate_",indicator,
235: "=(",indicator,"/lag(",indicator,"n=1)-1)/(Year-lag(Year, n=1))
236: )")))
237:   # "=(",indicator,"-lag(",indicator,"n=1))/(Year-lag(Year, n=1))/(",indicator,"+lag(",indicator,"n=1))*2
238:
239: } # 指標毎の処理2
240:
241: df_Graph <- df_Graph %>% ungroup() %>% arrange(SCENARIO,Country,Year)
242: # df_Graph <- df_Graph %>% ungroup() %>% group_by(SCENARIO,REGION) %>%
  arrange(SCENARIO,Country,Year)
243: # View(df_Graph)
244: write_csv(df_Graph, "../df_Graph_written.csv")
245:
246: #Summary -----
247:
248: indicators <-
  c('ChangeRate_Energy_Intensity','ChangeRate_Carbon_Intensity','ChangeRate_Electricity_Rate_Total',
249:   'ChangeRate_Electricity_Rate_Ind','ChangeRate_Electricity_Rate_Tra',
250:   'ChangeRate_Electricity_Rate_Res','ChangeRate_Electricity_Rate_Com',
251:   'Energy_Intensity_scaled','Carbon_Intensity_scaled','Electricity_Rate_Total',
252:   'Electricity_Rate_Ind','Electricity_Rate_Tra','Electricity_Rate_Res','Electricity_Rate_Com')
253: # indicators <- c('Energy_Intensity_scaled','ChangeRate_Energy_Intensity','ChangeRateBY_Energy_Intensity',
254: #   'Carbon_Intensity_scaled','ChangeRate_Carbon_Intensity','ChangeRateBY_Carbon_Intensity',
255: #   'Electricity_Rate_Total_scaled','ChangeRate_Electricity_Rate_Total','ChangeRateBY_Electricity_Rate_Total',
256: #   'Electricity_Rate_Total','POP_IEA','GDP_Capita')
257:
258: df_indicator <- df_Graph %>% select(one_of(Titlerow3),one_of(indicators)
259: ) %>% group_by(SCENARIO)

```

```

260: write_csv(df_indicator, "~/df_indicator_written.csv") # Year, Country, REGION入りのデータで保存
261:
262: df_summary <- df_indicator %>% select(-c(Year, Country, REGION))
263: ) %>% group_by(SCENARIO)
264: ) %>% summarise_at(vars(everything()),
265:   funs(length, n_distinct, min(., na.rm=T), median(., na.rm=T),
266:     max(., na.rm=T), mean(., na.rm=T), sd(., na.rm=T),
267:     'q5%'=quantile(., probs=0.05, na.rm=T),
268:     'q95%'=quantile(., probs=0.95, na.rm=T), )
269: ) # %>% arrange(colnames(df_summary ))
270:
271: # df_summary <- df_summary %>% mutate(item=names)
272: # colnames(df_summary) <- df_summary[1,]
273:
274:
275: df_summary_ChangeRate <- df_summary %>% select(SCENARIO, starts_with("ChangeRate_"))
276: df_summary_ChangeRate <- as.data.frame(t(df_summary_ChangeRate))
277: df_summary <- as.data.frame(t(df_summary))
278: write_csv(df_summary, "~/df_summary_written.csv", row.names=T)
279: write_csv(df_summary_ChangeRate, "~/df_summary_ChangeRate_written.csv", row.names=T)
280:
281:
282: #Graph output -----
283: for (dummyloop in 1) { # グラフ出力 for (dummyloop in 1) while (0)
284:
285:   # 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
286:   x_names <- c(rep('Year',length(indicators)),
287:     rep('GDP_Capita',length(indicators))
288:     ) # rep('REGION',length(indicators)),
289:   y_names <- c(rep(indicators,2)) #3
290:   scenario_color <- c('#3366CC', '#66AA00', '#0099C6', '#DD4477', '#BB2E2E', '#990099', '#651067',
291:     '#22AA99')
292:   axis_cutoff_percentile <- 0.005 # 軸の表示において切り捨てる分位範囲 (0.005: 両端5% cutoff)
293:   axis_range <- function(vec_indicator, cutoff_percentile) {
294:     axis_range_return <- c(quantile(na.omit(vec_indicator), cutoff_percentile),
295:       quantile(na.omit(vec_indicator), (1-cutoff_percentile))
296:     ) %>% as.numeric()
297:     return(axis_range_return)
298:   }
299:
300:   # scenarionames <- levels(df_Graph$SCENARIO) # c('Baseline','2C','1.5C','2.5C','WB2C') # 'Historical'
301:   scenarionames <- c('Multi')
302:   for (scenarioname in scenarionames) {
303:     if (scenarioname=='Multi') {
304:       df_Graph_plot <- df_Graph
305:     } else if (scenarioname=='Historical') {
306:       df_Graph_plot <- df_Graph %>% filter(SCENARIO=='Historical')
307:     } else {
308:       df_Graph_plot <- rbind(filter(df_Graph, SCENARIO==scenarioname),
309:         filter(df_Graph, SCENARIO=='Historical'))
310:     }
311:
312:     while (0) { # for (dummyloop in 1) { # XY散布図 by 17地域 bk
313:       pdf(file=paste("~/",scenarioname,"_XY.pdf", sep=""))
314:       for (num in 1:length(x_names)) {
315:         g <- eval(parse(text=paste0(
316:           "ggplot(df_Graph_plot, aes(x=",x_names[num],",y=",y_names[num],
317:             ",color=REGION, shape=SCENARIO)) +
318:             geom_line() +
319:             geom_point() +
320:             # scale_colour_gdocs() +
321:             scale_shape_manual(values=c(19,19,21,21,21,21,21,21))) # Historical2本
322:         plot(g)
323:         filename <- paste(scenarioname,num,"_",x_names[num],"-",y_names[num], sep="")
324:         # ggsave(file=paste("~/png/",filename,".png", sep=""), width=5, height=4, dpi=100)
325:       }
326:     }
327:   }

```

```

325:   dev.off()
326: } # XY散布図 by 17地域 bk
327:
328: while (0) { # 箱ヒゲ図 地域別
329:   pdf(file=paste("../",scenarioname,"_boxplot_Region.pdf", sep=""))
330:   for (indicator in indicators) {
331:     g <- eval(parse(text=paste0(
332:       "ggplot(df_Graph_plot, aes(x=REGION, y=", indicator, ", color=SCENARIO)) +
333:       geom_boxplot() +
334:       # geom_jitter(shape=20, position=position_dodge(0.8)) +
335:       scale_colour_gdocs() ")))
336:     plot(g)
337:     filename <- paste(scenarioname, "_", "boxplot_Region_", indicator, sep="")
338:     # ggsave(file=paste("../png/", filename, ".png", sep=""), width=5, height=4, dpi=100)
339:   }
340:   dev.off()
341: } # 箱ヒゲ図 地域別
342:
343: for (dummyloop in 1) { # 箱ヒゲ図 全世界
344:   pdf(file=paste("../",scenarioname,"_boxplot_World.pdf", sep=""))
345:   for (indicator in indicators) {
346:     g <- eval(parse(text=paste0(
347:       "ggplot(df_Graph_plot, aes(x=SCENARIO, y=", indicator, ", color=SCENARIO)) +
348:       geom_boxplot() +
349:       # geom_jitter(shape=20, position=position_dodge(0.8)) + # 箱ヒゲに点を重ねる
350:       stat_boxplot(geom='errorbar', width=0.3) + # ヒゲ先端の横線
351:       scale_colour_gdocs() ")))
352:     plot(g)
353:     filename <- paste(scenarioname, "_", "boxplot_World_", indicator, sep="")
354:     # ggsave(file=paste("../png/", filename, ".png", sep=""), width=5, height=4, dpi=100)
355:
356:     vec_indicator <- eval(parse(text=paste0("df_Graph_plot$", indicator)))
357:     axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
358:     g <- eval(parse(text=paste0(
359:       "ggplot(df_Graph_plot, aes(x=SCENARIO, y=", indicator, ", color=SCENARIO)) +
360:       geom_boxplot() +
361:       ylim("axis_range_value[1]", "axis_range_value[2]", " +
362:       stat_boxplot(geom='errorbar', width=0.3) + # ヒゲ先端の横線
363:       scale_colour_gdocs() ")))
364:     plot(g)
365:     filename <- paste(scenarioname, "_", "boxplot_World_ylim_", indicator, sep="")
366:     # ggsave(file=paste("../png/", filename, ".png", sep=""), width=5, height=4, dpi=100)
367:   }
368:   dev.off()
369: } # 箱ヒゲ図 全世界
370:
371: for (dummyloop in 1) { # 頻度分布
372:   pdf(file=paste("../",scenarioname,"_histogram.pdf", sep=""))
373:   for (indicator in indicators) {
374:     g <- eval(parse(text=paste0(
375:       "ggplot(df_Graph_plot, aes(x=", indicator, ", color=SCENARIO)) +
376:       # geom_histogram(bins=50) + # 積み上げ
377:       # geom_histogram(bins=50, position='identity', alpha=0.3) + # 透過重ね
378:       geom_histogram(bins=50, position='dodge', alpha=0) + # 隣接バー
379:       ylab('Count of Region-Year') +
380:       scale_colour_gdocs() ")))
381:     plot(g)
382:     filename <- paste(scenarioname, "_", "histogram_", indicator, sep="")
383:     # ggsave(file=paste("../png/", filename, ".png", sep=""), width=5, height=4, dpi=100)
384:
385:     vec_indicator <- eval(parse(text=paste0("df_Graph_plot$", indicator)))
386:     axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
387:     g <- eval(parse(text=paste0(
388:       "ggplot(df_Graph_plot, aes(x=", indicator, ", color=SCENARIO)) +
389:       geom_histogram(bins=50, position='dodge', alpha=0) + # 隣接バー
390:       ylab('Count of Region-Year') +

```

```

391:     xlim("",axis_range_value[1], "", "",axis_range_value[2], "") +
392:     scale_colour_gdocs() ""))
393: plot(g)
394: filename <- paste(scenarioname,"_", "histogram_xlim_",indicator, sep="")
395: # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
396: }
397: dev.off()
398: } # 頻度分布
399:
400: for (dummyloop in 1) { # 確率密度分布
401:   pdf(file=paste("./",scenarioname,"_density.pdf", sep=""))
402:   for (indicator in indicators) {
403:     g <- eval(parse(text=paste0(
404:       "ggplot(df_Graph_plot, aes(x=",indicator, ",color=SCENARIO)) +
405:       geom_density(size=0.7) +
406:       scale_colour_gdocs() +
407:       # xlim(-0.2,0.2) +
408:       ylab('Density (Counts scaled to 1) of Region-Year')"))
409:     plot(g)
410:     filename <- paste(scenarioname,"_", "density_",indicator, sep="")
411:     # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
412:
413:     vec_indicator <- eval(parse(text=paste0("df_Graph_plot$",indicator)))
414:     axis_range_value <- axis.range(vec_indicator, axis_cutoff_percentile)
415:     g <- eval(parse(text=paste0(
416:       "ggplot(df_Graph_plot, aes(x=",indicator, ",color=SCENARIO)) +
417:       geom_density(size=0.7) +
418:       scale_colour_gdocs() +
419:       xlim("",axis_range_value[1], "", "",axis_range_value[2], "") +
420:       # xlim(-0.2,0.2) +
421:       ylab('Density (Counts scaled to 1) of Region-Year')"))
422:     plot(g)
423:     filename <- paste(scenarioname,"_", "density_xlim_",indicator, sep="")
424:     # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
425:
426:   }
427:   dev.off()
428: } # 確率密度分布
429:
430: for (dummyloop in 1) { # XY散布図 by 17地域 vs 17地域
431:   library(RColorBrewer)
432:   scenario_color <- c(brewer.pal(5,"Dark2"),brewer.pal(8,"Accent"),brewer.pal(4,"Set1"))
433:   # df_Graph_plotXY <- data.frame()
434:   df_Graph_plotXY <- df_Graph_plot %>% filter(SCENARIO!='Historical')
435:   df_Graph_plotXY_His <- df_Graph_plotXY %>% filter(SCENARIO=='Historical_R17')
436:
437:   write_csv(df_Graph_plotXY, "./df_Graph_plotXY_written.csv")
438:   write_csv(df_Graph_plotXY_His, "./df_Graph_plotXY_His_written.csv")
439:
440:   pdf(file=paste("./",scenarioname,"_XY_R17.pdf", sep=""))
441:   for (num in 1:length(x_names)) { #num
442:     g <- eval(parse(text=paste0(
443:       "ggplot(df_Graph_plotXY, aes(x=",x_names[num],",y=",y_names[num],
444:       ",color=REGION, shape=SCENARIO)) +
445:       geom_point() +
446:       geom_line() +
447:       scale_color_manual(values=c(rep(scenario_color,3))) +
448:       scale_shape_manual(values=c(19,21,22,23,24,25,1))")) # SCENARIO数
449:     plot(g)
450:     filename <- paste(scenarioname,num,"_",x_names[num],"-",y_names[num], sep="")
451:     # ggsave(file=paste("./png/R17",filename,".png", sep=""), width=5, height=4, dpi=100)
452:
453:     small <- 0.001
454:     x_axis_min <- min(eval(parse(text=paste0("df_Graph_plotXY$",x_names[num]))), na.rm=T)*(1-small)
455:     y_axis_min <- min(eval(parse(text=paste0("df_Graph_plotXY$",y_names[num]))), na.rm=T)*(1-small)
456:     x_axis_max <- max(eval(parse(text=paste0("df_Graph_plotXY$",x_names[num]))), na.rm=T)*(1+small)

```

```

457:   y_axis_max <- max(eval(parse(text=paste0("df_Graph_plotXY$", y_names[num]))), na.rm=T)*(1+small)
458:
459:   g <- eval(parse(text=paste0(
460:     "ggplot(df_Graph_plotXY_His, aes(x=", x_names[num], ", y=", y_names[num],
461:     ", color=REGION, shape=SCENARIO)) +
462:     geom_point() +
463:     geom_line() +
464:     xlim("x_axis_min", "x_axis_max") +
465:     ylim("y_axis_min", "y_axis_max") +
466:     scale_color_manual(values=c(rep(scenario_color, 3))) +
467:     scale_shape_manual(values=c(19, 21, 22, 23, 24, 25, 1)))) # SCENARIO数
468:   plot(g)
469: } #num
470: dev.off()
471: } # XY散布図 by 17地域 vs 17地域
472:
473: while (0) { # XY散布図 by 国別 for (dummyloop in 1)
474:   df_Graph_plot <- df_Graph_plot %>% ungroup() %>% group_by(Country, SCENARIO)
475:   pdf(file=paste("./", scenarioname, "_XY_Country.pdf", sep=""))
476:   for (num in 1:length(x_names)) {
477:     g <- eval(parse(text=paste0(
478:       "ggplot(df_Graph_plot, aes(x=", x_names[num], ", y=", y_names[num],
479:       ", color=REGION, shape=SCENARIO)) +
480:       geom_line() +
481:       geom_point() +
482:       scale_colour_gdocs() +
483:       # theme(legend.position='none') +
484:       scale_shape_manual(values=c(19, 19, 21, 21, 21, 21, 21))))
485:     plot(g)
486:     filename <- paste(scenarioname, "_", num, "_", x_names[num], "_", y_names[num], "_CN", sep="")
487:     # ggsave(file=paste("./png/", filename, ".png", sep=""), width=5, height=4, dpi=100)
488:   }
489:   dev.off()
490: } # XY散布図 by 国別
491:
492: while (0) { # 特定パターンのみ の確率密度分布 for (dummyloop in 1)
493:   df_Graph_developed <- df_Graph_plot %>% filter(REGION %in% c('CAN', 'CIS', 'JPN', 'USA', 'XE25', 'XER')) # 先進国
494:   df_Graph_developing <- anti_join(df_Graph_plot, df_Graph_developed) # 先進国以外
495:   df_Graph_filtered <- df_Graph_developing # グラフ対象
496:
497:   # library(outliers)
498:   # indicator <- c('ChangeRate_Carbon_Intensity')
499:   # x_range_all <- df_Graph_filtered %>% select(all_of(indicator)) %>% drop_na()
500:   # x_range_outliered <- df_Graph_filtered %>% select(all_of(indicator))
501:   # %>% rm.outlier(fill=F, median=F, opposite=F)
502:   # %>% drop_na() # 外れ値を除外
503:   # anti_join(x_range_all, x_range_outliered)
504:   # semi_join(x_range_all, x_range_outliered)
505:
506:   pdf(file=paste("./", scenarioname, "_density_filtered.pdf", sep=""))
507:   for (indicator in indicators) {
508:     g <- eval(parse(text=paste0(
509:       "ggplot(df_Graph_filtered, aes(x=", indicator, ", color=SCENARIO)) +
510:       geom_density(size=0.7) +
511:       ylab('Density (Counts scaled to 1) of Region-Year') +
512:       scale_colour_gdocs()))
513:     plot(g)
514:     filename <- paste(scenarioname, "_", "density_filtered_", indicator, sep="")
515:     # ggsave(file=paste("./filtered/", filename, ".png", sep=""), width=5, height=4, dpi=100)
516:
517:     # 範囲指定のグラフ
518:     vec_indicator <- eval(parse(text=paste0("df_Graph_plot$", indicator)))
519:     axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
520:     g <- eval(parse(text=paste0(
521:       "ggplot(df_Graph_filtered, aes(x=", indicator, ", color=SCENARIO)) +

```



```
522:     geom_density(size=0.7) +
523:     xlim("",axis_range_value[1], "", "",axis_range_value[2], "") + #
524:     ylab('Density (Counts scaled to 1) of Region-Year') +
525:     scale_colour_gdocs()))))
526:   plot(g)
527:   # ggsave(file=paste("./filtered/",filename,"_xlim.png", sep=""), width=5, height=4, dpi=100)
528: }
529:   dev.off()
530: } # 特定パターンの方の確率密度分布
531: } # scenarioname loop
532: } # グラフ出力
533: }
534: }
535:
```