```
# Packages -----------------------------------------------------------
library(ggplot2)
library(ggthemes)
library(tidyverse)

root <- 'C:/_Nishimoto/R/WBAL_R02/'
Titlerow1 <- c('MODEL','SCENARIO','REGION','VARIABLE','UNIT')
Titlerow2 <- c('REGION','Country','VARIABLE','SCENARIO')
Titlerow3 <- c('SCENARIO','Country','REGION','Year')
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）
BaseYear <- 2010  # %>% as.numeric()  # 基準年値
Sample_Country <- c('Former Soviet Union','Former Yugoslavia','South Sudan','Bosnia and Herzegovina')  # GDP(2010)が無い国
Interpolate_NA <- 'fill'   # 'fill_latest_or_first_existing_value'
Year5 <- c(1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015,
           2020, 2025, 2030, 2035, 2040, 2045, 2050, 2055, 2060, 2065,
           2070, 2075, 2080, 2085, 2090, 2095, 2100) # %>% as.character()
# Year==0 as Base-Year

# Past -----------------------------------------------------------
setwd(paste(root,"2_data/REF", sep=""))

while (0) {
  # 単位の連想配列＞ファイル名にマッチさせる予定
  df_unit <- read.delim(file="./unit.txt", header=T)
  # view(df_unit)
  Unit_of_Var         <- df_unit$unit
  names(Unit_of_Var) <- df_unit$filename
  # Unit_of_Var[names(Unit_of_Var)]
}

# 国コードの連想配列
df_CC <- read.delim(file="./CC.txt", header=T)
df_CC <- rename(df_CC, 'Country'='IEA国名')
# view(df_CC)
Region_Code         <- df_CC$AIM17
names(Region_Code) <- df_CC$Country
# Region_Code[names(Region_Code)]

# タイトル行（ダミー）の作成
df_past <- read_csv("./POP_IEA.csv")
df_past <- df_past  %>% mutate('REGION'='region', 'VARIABLE'='variable'
                   ) %>% rename('Country'='TIME') # 'TIME' OR 'X1'
df_past <- df_past[1,c(ncol(df_past),ncol(df_past)-1,1:(ncol(df_past)-2))] # 列の入替
# View(df_past)

# 1ファイル毎に追加
files  <- list.files()    # 指定ディレクトリのファイル一覧を代入
for (file_name in files) {
  if ( regexpr('¥¥.csv$', file_name) < 0 ) { next }
  d  <- read_csv(file_name)  # ファイルを仮変数に読み込む
  file_name <- gsub(".csv", "", file_name)
  d  <- d %>% rename(REGION=TIME
        ) %>% mutate(VARIABLE=file_name)

  # 国コード付与
  d  <- d   %>% mutate(AIM17=Region_Code[d$REGION]
            ) %>% rename('Country'='REGION'
            ) %>% rename('REGION'='AIM17'
            ) %>% drop_na('REGION')  # 国コードのない行は無視
  #  d <- d[1,c(ncol(d),1:(ncol(d)-1))] # 列の入替
  df_past <- rbind(df_past, d)
}
setwd(paste(root,"4_output/", sep=""))

df_past <- df_past %>% filter(REGION!='region')  # ダミー行のデータを削除
df_past <- df_past %>% mutate(SCENARIO='Historical')  # 書式を揃える
df_past <- df_past %>% mutate(Country = str_replace_all(Country,
                       pattern = c("Memo.: "="", "Memo: "="", " .if no detail."="")))
write_csv(df_past, "./df_past_written_everyYear.csv") # VARIABLE REGION Country

df_past_long <- df_past %>% gather(key='Year', value='Value', -all_of(Titlerow2))
df_past_long$Year  <- as.numeric(df_past_long$Year)
df_past_long$Value <- as.numeric(df_past_long$Value)  # NA warning ＞ 確認済

# Fill past ---------------------------------------------------------
# 基準年データがない国の処理 for (dummyloop in 1) {

df_past_long <- df_past_long %>% group_by(VARIABLE,Country
      ) %>% arrange(VARIABLE, Country, Year
      ) %>% mutate(Value2=Value
      ) %>% fill(Value2, .direction='down' # 前年値を優先
      ) %>% fill(Value2, .direction='up'  # 前年値がなければ後年値
      ) %>% mutate(SCENARIO2=if_else(is.na(Value), Interpolate_NA, SCENARIO)
      ) %>% ungroup()

df_past_BaseYear <- df_past_long %>% filter(Year==BaseYear
      ) %>% mutate(Year=0)

while (0) { # PDF出力
  pdf(file=paste("./","past_filled.pdf", sep=""))  # PDF出力開始
  for (y_name in unique(df_past_long$VARIABLE)) { # 補完値の確認出力 by サンプル国
    df_Graph_past <- df_past_long %>% filter(Country %in% Sample_Country
                                ) %>% filter(VARIABLE==y_name)
    g <- ggplot(df_Graph_past, aes(x=Year,y=Value2,
                                   color=Country, shape=SCENARIO2)) +
      geom_point() +
      # theme(legend.position='none') +
      ylab(y_name) +
      scale_shape_manual(values=c(24,19))
    plot(g)
    filename <- paste("Interpolated_",y_name,"_",Interpolate_NA, sep="")
    # ggsave(file=paste("./png/",filename,".png", sep=""))
  }
  dev.off() # PDF出力終了
} # PDF出力

# 基準年データがない国の処理 }

write_csv(df_past_long, "./df_past_long_written_everyYear.csv") # VARIABLE REGION Country
write_csv(df_past, "./df_past_written_everyYear.csv") # VARIABLE REGION Country

# Future ---------------------------------------------------------
# while (0) {  # 将来シナリオの読込
df_future <- read_csv(paste(root,"2_data/REF2/","IAMCTemplate.csv", sep=""))
# View(df_future)

df_future <- df_future %>% select(-c('MODEL','UNIT')
# ) %>% filter(SCENARIO == scenarioname # rbind前にシナリオを絞る場合
) %>% filter(!REGION %in% c('ASIA2', 'World')
) %>% mutate(Country = REGION)  # 書式を揃える # 国名=地域名

df_future <- df_future %>% mutate('0'=as.numeric(df_future$'2010')) # 基準年値をコピーしておく

# IAMCTemplete(future) の名前を IEA(past) に揃える@ '|'対策  # recode でも出来るらしい
df_future <- df_future %>% mutate(VARIABLE = str_replace_all(VARIABLE, pattern = c(
  'GDP.MER' = 'GDP_IEA',
  'Population' = 'POP_IEA',
  'Primary Energy' = 'TES_Total',
  'Emissions.CO2.Energy' = 'CO2_fuel_Total',
  'Final Energy.Industry.Electricity' = 'TFC_Elec_Ind',
  'Final Energy.Transportation.Electricity' = 'TFC_Elec_Tra',
  'Final Energy.Residential.Electricity' = 'TFC_Elec_Res',
  'Final Energy.Commercial.Electricity' = 'TFC_Elec_Com',
  'Final Energy.Electricity' = 'TFC_Elec_Total',
  'Final Energy.Industry' = 'TFC_Total_Ind',
  'Final Energy.Transportation' = 'TFC_Total_Tra',
  'Final Energy.Residential' = 'TFC_Total_Res',
  'Final Energy.Commercial' = 'TFC_Total_Com',
  'Final Energy' = 'TFC_Total_Total')))  # 順番注意／.最長マッチ
# View(df_future)
write_csv(df_future, "./df_future_pre_written.csv")
# ここで、上記以外のデータは捨てる
df_future <- df_future %>% filter(VARIABLE %in%
              c('GDP_IEA','POP_IEA','TES_Total','CO2_fuel_Total',
                'TFC_Total_Total','TFC_Total_Ind','TFC_Total_Tra','TFC_Total_Res','TFC_Total_Com',
                'TFC_Elec_Total','TFC_Elec_Ind','TFC_Elec_Tra','TFC_Elec_Res','TFC_Elec_Com'))
write_csv(df_future, "./df_future_written.csv")
# }  # 将来シナリオの読込

# Connect Past & Future -----------------------------------------------
df_long_past <- df_past_long %>% rbind(df_past_BaseYear)
df_long_past <- df_long_past %>% select(-Value2, -SCENARIO2
                      ) %>% filter(Year %in% c(Year5, 0))  # 各年から5年置きに
df_long_future <- gather(df_future, key=Year, value=Value, -all_of(Titlerow2))
df_long_future$Year  <- as.numeric(df_long_future$Year)
df_long <- rbind(df_long_past, df_long_future)
df_long$Year  <- as.numeric(df_long$Year)
df_long$Value <- as.numeric(df_long$Value)  # NA warning > 確認済
write_csv(df_long, "./df_long_written.csv")

# Aggregation to Region -----------------------------------------------
for (dummyloop in 1) {  # 地域集約 # while (0)
  df_long_agg <- aggregate(Value~VARIABLE+REGION+SCENARIO+Year, df_long, sum) # 集約対象=Country
  df_long_agg <- df_long_agg %>% mutate(Country=REGION)
  # df_long <- df_long_agg  # 過去17地域、将来17地域で出力する場合
  for (dummyloop in 1) {  # 過去17地域と過去約2百数ヶ国を併記する場合
    df_long_agg <- df_long_agg %>% filter(SCENARIO=='Historical'
                          ) %>% mutate(SCENARIO='Historical_R17')
    df_long <- df_long %>% rbind(df_long_agg)
  } # 併記する場合
  write_csv(df_long_agg, "./df_long_agg_written.csv")
}  # 地域集約


# Table format and Indicator  -----------------------------------------------
# 指標の処理  # Variable_Names_for_Indicators df_vni <- indicator, numerator, denominator
df_vni <- matrix(c(
  'GDP_Capita',    'GDP_IEA', 'POP_IEA',
  'Energy_Intensity',    'TES_Total', 'GDP_IEA',
  'Carbon_Intensity',    'CO2_fuel_Total', 'TES_Total',
  'Electricity_Rate_Total',    'TFC_Elec_Total', 'TFC_Total_Total',
  'Electricity_Rate_Ind',    'TFC_Elec_Ind',    'TFC_Total_Ind',
  'Electricity_Rate_Tra',    'TFC_Elec_Tra',    'TFC_Total_Tra',
  'Electricity_Rate_Res',    'TFC_Elec_Res',    'TFC_Total_Res',
  'Electricity_Rate_Com',    'TFC_Elec_Com',    'TFC_Total_Com'),
  ncol=8, nrow=3)

for (dummyloop in 1) {  # 国名のみのダミー列の作成
  df_Graph <- df_long %>% select(c('Country')) %>% arrange(Country) %>% distinct()
}  # ダミー列の作成

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))
  indicator   <- df_vni[1,i]
  numerator   <- df_vni[2,i]
  denominator <- df_vni[3,i]

  for (variable_name in c(numerator, denominator)) {
    df_toJoin <- df_long %>% filter(VARIABLE==variable_name
                     ) %>% select(-c('VARIABLE')
                     ) %>% arrange(Year)
    df_toJoin <- eval(parse(text=paste0("df_toJoin %>% rename(",variable_name,"=Value)")))
    # View(df_toJoin)
    df_Graph <- df_Graph %>% full_join(df_toJoin)
  }
  df_Graph <- df_Graph %>% drop_na('REGION','Year')  # ダミー列のデータを削除
  df_Graph <- eval(parse(text=paste0(
             "df_Graph %>% mutate(",indicator,"=",numerator,"/",denominator,")"))) # 指標の算出

  for (dummyloop in 1) { # 基準年値をdf_Graphに追加する(0年値として追加)

    df_Graph <- df_Graph %>% group_by(SCENARIO,Country) %>% arrange(SCENARIO,Country,Year)
    df_Graph<- eval(parse(text=paste0(
      "df_Graph %>% mutate(",indicator,"_scaled=",indicator,"/",indicator,"[Year==0])"
    )))                     # indicator_scaled = I(t)/I(t=BaseYear)
    df_Graph <- df_Graph %>% ungroup()

  } # 基準年値をdf_Graphに追加する

} # 指標毎の処理1
df_Graph$SCENARIO <- factor(df_Graph$SCENARIO,
                             levels=c('Historical','Historical_R17','Baseline','2.5C','2C','1.5C','WB2C'))
df_Graph <- df_Graph %>% filter(Year!=0) %>% group_by(SCENARIO,Country) %>% arrange(SCENARIO,Country,Year)
write_csv(df_Graph, "./df_Graph_afterfulljoin_written.csv")

# Change rate -------------------------------------------------------
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))

  indicator   <- df_vni[1,i]

  # 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))
  df_Graph <- eval(parse(text=paste0(
        "df_Graph %>%  mutate(ChangeRateBY_",indicator,
        "=(",indicator,"_scaled-lag(",indicator,"_scaled, n=1))/(Year-lag(Year, n=1))
                ) %>% mutate(ChangeRate_",indicator,
        "=(",indicator,"/lag(",indicator,",n=1)-1)/(Year-lag(Year, n=1))
                )")))
  #  "=(",indicator,"-lag(",indicator,",n=1))/(Year-lag(Year, n=1))/(",indicator,"+lag(",indicator,",n=1))*2

} # 指標毎の処理2

df_Graph <- df_Graph %>% ungroup() %>% arrange(SCENARIO,Country,Year)
# df_Graph <- df_Graph %>% ungroup() %>% group_by(SCENARIO,REGION) %>% arrange(SCENARIO,Country,Year)
# View(df_Graph)
write_csv(df_Graph, "./df_Graph_written.csv")

#Summary -------------------------------------------------------

indicators <- c('ChangeRate_Energy_Intensity','ChangeRate_Carbon_Intensity','ChangeRate_Electricity_Rate_Total',
                'ChangeRate_Electricity_Rate_Ind','ChangeRate_Electricity_Rate_Tra',
                'ChangeRate_Electricity_Rate_Res','ChangeRate_Electricity_Rate_Com',
                'Energy_Intensity_scaled','Carbon_Intensity_scaled','Electricity_Rate_Total',
                'Electricity_Rate_Ind','Electricity_Rate_Tra','Electricity_Rate_Res','Electricity_Rate_Com')
# indicators <- c('Energy_Intensity_scaled','ChangeRate_Energy_Intensity','ChangeRateBY_Energy_Intensity',
#                 'Carbon_Intensity_scaled','ChangeRate_Carbon_Intensity','ChangeRateBY_Carbon_Intensity',
# 'Electricity_Rate_Total_scaled','ChangeRate_Electricity_Rate_Total','ChangeRateBY_Electricity_Rate_Total',
#                 'Electricity_Rate_Total','POP_IEA','GDP_Capita')

df_indicator <- df_Graph %>% select(one_of(Titlerow3),one_of(indicators)
                      ) %>% group_by(SCENARIO)
```

```
write_csv(df_indicator, "./df_indicator_written.csv")  # Year, Country, REGION入りのデータで保存

df_summary <- df_indicator %>% select(-c(Year, Country, REGION)
  ) %>% group_by(SCENARIO
  ) %>% summarise_at(vars(everything()),
                     funs(length, n_distinct, min(., na.rm=T), median(., na.rm=T),
                          max(., na.rm=T), mean(., na.rm=T), sd(., na.rm=T),
                          'q5%'=quantile(., probs=0.05, na.rm=T),
                          'q95%'=quantile(., probs=0.95, na.rm=T), )
  ) # %>% arrange(colnames(df_summary ))

# df_summary <- df_summary  %>% mutate(item=names)
# colnames(df_summary) <- df_summary[1,]


df_summary_ChangeRate <- df_summary %>% select(SCENARIO, starts_with("ChangeRate_"))
df_summary_ChangeRate <- as.data.frame(t(df_summary_ChangeRate))
df_summary <- as.data.frame(t(df_summary))
write.csv(df_summary, "./df_summary_written.csv", row.names=T)
write.csv(df_summary_ChangeRate, "./df_summary_ChangeRate_written.csv", row.names=T)


#Graph output ------------------------------------------------
for (dummyloop in 1) {  # グラフ出力 for (dummyloop in 1) while (0)

  # 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
  x_names <- c(rep('Year',length(indicators)),
               rep('GDP_Capita',length(indicators))
               ) # rep('REGION',length(indicators)),
  y_names <- c(rep(indicators,2)) #3
  scenario_color <- c('#3366CC', '#66AA00', '#0099C6', '#DD4477', '#BB2E2E', '#990099', '#651067', '#22AA99')
  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)
  axis_range <- function(vec_indicator, cutoff_percentile) {
    axis_range_return <- c(quantile(na.omit(vec_indicator), cutoff_percentile),
                           quantile(na.omit(vec_indicator), (1-cutoff_percentile))
    ) %>% as.numeric()
    return(axis_range_return)
  }

  # scenarionames <- levels(df_Graph$SCENARIO)    # c('Baseline','2C','1.5C','2.5C','WB2C') # 'Historical'
  scenarionames <- c('Multi')
  for (scenarioname in scenarionames) {
    if (scenarioname=='Multi') {
        df_Graph_plot <- df_Graph
    } else if (scenarioname=='Historical') {
        df_Graph_plot <- df_Graph %>% filter(SCENARIO=='Historical')
    } else {
        df_Graph_plot <- rbind(filter(df_Graph, SCENARIO==scenarioname),
                               filter(df_Graph, SCENARIO=='Historical'))
    }

    while (0) { # for (dummyloop in 1) { # XY散布図 by 17地域 bk
        pdf(file=paste("./",scenarioname,"_XY.pdf", sep=""))
        for (num in 1:length(x_names)) {
          g <- eval(parse(text=paste0(
            "ggplot(df_Graph_plot, aes(x=",x_names[num],",y=",y_names[num],
            ",color=REGION, shape=SCENARIO)) +
              geom_line() +
              geom_point() +
            # scale_colour_gdocs() +
              scale_shape_manual(values=c(19,19,21,21,21,21,21))"))) # Historical2本
          plot(g)
          filename <- paste(scenarioname,num,"_",x_names[num],"-",y_names[num], sep="")
          # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
        }
```

```
      dev.off()
  } # XY散布図 by 17地域 bk

  while (0) { # 箱ヒゲ図　地域別
    pdf(file=paste("./",scenarioname,"_boxplot_Region.pdf", sep=""))
    for (indicator in indicators) {
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=REGION ,y=",indicator, ", color=SCENARIO)) +
          geom_boxplot() +
        # geom_jitter(shape=20, position=position_dodge(0.8)) +
          scale_colour_gdocs() ")))
      plot(g)
      filename <- paste(scenarioname,"_","boxplot_Region_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
    }
    dev.off()
  } # 箱ヒゲ図　地域別

  for (dummyloop in 1) { # 箱ヒゲ図　全世界
    pdf(file=paste("./",scenarioname,"_boxplot_World.pdf", sep=""))
    for (indicator in indicators) {
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=SCENARIO, y=",indicator, ", color=SCENARIO)) +
          geom_boxplot() +
        # geom_jitter(shape=20, position=position_dodge(0.8)) +  # 箱ヒゲに点を重ねる
          stat_boxplot(geom='errorbar', width=0.3) + # ヒゲ先端の横線
          scale_colour_gdocs() ")))
      plot(g)
      filename <- paste(scenarioname,"_","boxplot_World_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)

      vec_indicator <- eval(parse(text=paste0("df_Graph_plot$",indicator)))
      axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=SCENARIO, y=",indicator, ", color=SCENARIO)) +
          geom_boxplot() +
          ylim(",axis_range_value[1], ", ",axis_range_value[2], ") +
          stat_boxplot(geom='errorbar', width=0.3) + # ヒゲ先端の横線
          scale_colour_gdocs() ")))
      plot(g)
      filename <- paste(scenarioname,"_","boxplot_World_ylim_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
    }
    dev.off()
  } # 箱ヒゲ図　全世界

  for (dummyloop in 1) { # 頻度分布
    pdf(file=paste("./",scenarioname,"_histogram.pdf", sep=""))
    for (indicator in indicators) {
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=",indicator, ",color=SCENARIO)) +
        # geom_histogram(bins=50) + # 積み上げ
        # geom_histogram(bins=50, position='identity', alpha=0.3) + # 透過重ね
          geom_histogram(bins=50, position='dodge', alpha=0) + # 隣接バー
          ylab('Count of Region-Year') +
          scale_colour_gdocs() ")))
      plot(g)
      filename <- paste(scenarioname,"_","histogram_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)

      vec_indicator <- eval(parse(text=paste0("df_Graph_plot$",indicator)))
      axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=",indicator, ",color=SCENARIO)) +
         geom_histogram(bins=50, position='dodge', alpha=0) + # 隣接バー
          ylab('Count of Region-Year') +
          xlim(",axis_range_value[1], ", ",axis_range_value[2], ") +
          scale_colour_gdocs() ")))
      plot(g)
      filename <- paste(scenarioname,"_","histogram_xlim_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
    }
    dev.off()
  } # 頻度分布

  for (dummyloop in 1) { # 確率密度分布
    pdf(file=paste("./",scenarioname,"_density.pdf", sep=""))
    for (indicator in indicators) {
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=",indicator, ",color=SCENARIO)) +
          geom_density(size=0.7) +
          scale_colour_gdocs() +
        # xlim(-0.2,0.2) +
          ylab('Density (Counts scaled to 1) of Region-Year')")))
      plot(g)
      filename <- paste(scenarioname,"_","density_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)

      vec_indicator <- eval(parse(text=paste0("df_Graph_plot$",indicator)))
      axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plot, aes(x=",indicator, ",color=SCENARIO)) +
          geom_density(size=0.7) +
          scale_colour_gdocs() +
          xlim(",axis_range_value[1], ", ",axis_range_value[2], ") +
        # xlim(-0.2,0.2) +
          ylab('Density (Counts scaled to 1) of Region-Year')")))
      plot(g)
      filename <- paste(scenarioname,"_","density_xlim_",indicator, sep="")
      # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)

    }
    dev.off()
  } # 確率密度分布

  for (dummyloop in 1) { # XY散布図 by 17地域 vs 17地域
    library(RColorBrewer)
    scenario_color <- c(brewer.pal(5,"Dark2"),brewer.pal(8,"Accent"),brewer.pal(4,"Set1"))
    # df_Graph_plotXY <- data_frame()
    df_Graph_plotXY <- df_Graph_plot %>% filter(SCENARIO!='Historical')
    df_Graph_plotXY_His <- df_Graph_plotXY %>% filter(SCENARIO=='Historical_R17')

    write_csv(df_Graph_plotXY, "./df_Graph_plotXY_written.csv")
    write_csv(df_Graph_plotXY_His, "./df_Graph_plotXY_His_written.csv")

    pdf(file=paste("./",scenarioname,"_XY_R17.pdf", sep=""))
    for (num in 1:length(x_names)) { #num
      g <- eval(parse(text=paste0(
        "ggplot(df_Graph_plotXY, aes(x=",x_names[num],",y=",y_names[num],
        ",color=REGION, shape=SCENARIO)) +
            geom_point() +
            geom_line() +
            scale_color_manual(values=c(rep(scenario_color,3))) +
            scale_shape_manual(values=c(19,21,22,23,24,25,1))"))) # SCENARIO数
      plot(g)
      filename <- paste(scenarioname,num,"_",x_names[num],"-",y_names[num], sep="")
      # ggsave(file=paste("./png/R17",filename,".png", sep=""), width=5, height=4, dpi=100)

      small <- 0.001
      x_axis_min <- min(eval(parse(text=paste0("df_Graph_plotXY$",x_names[num]))), na.rm=T)*(1-small)
      y_axis_min <- min(eval(parse(text=paste0("df_Graph_plotXY$",y_names[num]))), na.rm=T)*(1-small)
      x_axis_max <- max(eval(parse(text=paste0("df_Graph_plotXY$",x_names[num]))), na.rm=T)*(1+small)
```

```
        y_axis_max <- max(eval(parse(text=paste0("df_Graph_plotXY$",y_names[num]))), na.rm=T)*(1+small)

        g <- eval(parse(text=paste0(
          "ggplot(df_Graph_plotXY_His, aes(x=",x_names[num],",y=",y_names[num],
          ",color=REGION, shape=SCENARIO)) +
              geom_point() +
              geom_line() +
              xlim(",x_axis_min, ", ",x_axis_max, ") +
              ylim(",y_axis_min, ", ",y_axis_max, ") +
              scale_color_manual(values=c(rep(scenario_color,3))) +
              scale_shape_manual(values=c(19,21,22,23,24,25,1))"))) # SCENARIO数
        plot(g)
     } #num
     dev.off()
    } # XY散布図 by 17地域 vs 17地域

    while (0) { # XY散布図 by 国別 for (dummyloop in 1)
      df_Graph_plot <- df_Graph_plot %>% ungroup() %>% group_by(Country,SCENARIO)
      pdf(file=paste("./",scenarioname,"_XY_Country.pdf", sep=""))
      for (num in 1:length(x_names)) {
        g <- eval(parse(text=paste0(
          "ggplot(df_Graph_plot, aes(x=",x_names[num],",y=",y_names[num],
          ",color=REGION, shape=SCENARIO)) +
            geom_line() +
            geom_point() +
            scale_colour_gdocs() +
          # theme(legend.position='none') +
            scale_shape_manual(values=c(19,19,21,21,21,21,21))")))
        plot(g)
        filename <- paste(scenarioname,"_",num,"_",x_names[num],"-",y_names[num],"_CN", sep="")
        # ggsave(file=paste("./png/",filename,".png", sep=""), width=5, height=4, dpi=100)
      }
      dev.off()
    } # XY散布図 by 国別

    while (0) { # 特定パターンのみの確率密度分布  for (dummyloop in 1)
      df_Graph_developed <- df_Graph_plot %>% filter(REGION %in% c('CAN','CIS','JPN','USA','XE25','XER')) # 先進国
      df_Graph_developing <- anti_join(df_Graph_plot, df_Graph_developed) # 先進国以外
      df_Graph_filtered <- df_Graph_developing # グラフ対象

      # library(outliers)
      # indicator <- c('ChangeRate_Carbon_Intensity')
      # x_range_all <- df_Graph_filtered %>% select(all_of(indicator)) %>% drop_na()
      # x_range_outliered <- df_Graph_filtered %>% select(all_of(indicator)
      #                                 ) %>% rm.outlier(fill=F, median=F, opposite=F
      #                                 ) %>% drop_na()  # 外れ値を除外
      # anti_join(x_range_all, x_range_outliered)
      # semi_join(x_range_all, x_range_outliered)

      pdf(file=paste("./",scenarioname,"_density_filtered.pdf", sep=""))
      for (indicator in indicators) {
        g <- eval(parse(text=paste0(
          "ggplot(df_Graph_filtered, aes(x=",indicator, ",color=SCENARIO)) +
            geom_density(size=0.7) +
            ylab('Density (Counts scaled to 1) of Region-Year') +
            scale_colour_gdocs()")))
        plot(g)
        filename <- paste(scenarioname,"_","density_filtered_",indicator, sep="")
        # ggsave(file=paste("./filtered/",filename,".png", sep=""), width=5, height=4, dpi=100)

        # 範囲指定のグラフ
        vec_indicator <- eval(parse(text=paste0("df_Graph_plot$",indicator)))
        axis_range_value <- axis_range(vec_indicator, axis_cutoff_percentile)
        g <- eval(parse(text=paste0(
          "ggplot(df_Graph_filtered, aes(x=",indicator, ",color=SCENARIO)) +
            geom_density(size=0.7) +
            xlim(",axis_range_value[1], ", ",axis_range_value[2], ") +  #
            ylab('Density (Counts scaled to 1) of Region-Year') +
            scale_colour_gdocs()")))
        plot(g)
        # ggsave(file=paste("./filtered/",filename,"_xlim.png", sep=""), width=5, height=4, dpi=100)

      }
      dev.off()
    } # 特定パターンのみの確率密度分布

  } # scenarioname loop
} # グラフ出力

```