Target 1 – GDP per capita growth

First, we load all required packages.

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(rnaturalearth)
library(rnaturalearthdata)
library(sf)
```

Here we import the three main datasets used for Target 1: the continent classifications, GDP per capita from the World Bank, and the NEET dataset (used later for context).

```
continents.according.to.our.world.in.data <- read.csv("~/Documents/GitHub/Data-Science-Coursework/conti
gdp.per.capita.worldbank <- read.csv("~/Documents/GitHub/Data-Science-Coursework/gdp-per-capita-worldba
youth.not.in.education.employment.training <- read.csv("~/Documents/GitHub/Data-Science-Coursework/youtl
```

This chunk cleans the continent data: we sort countries by continent, remove Antarctica, and drop the `Year` column so it is easier to join with the GDP dataset later.

```
Sorted_continents <- continents.according.to.our.world.in.data %>%
  arrange(Continent) %>%
  filter(Continent != "Antarctica")

Sorted_continents <- Sorted_continents[, -3]
```

We now clean the GDP per capita data: remove the first row (header row), rename columns, and convert `Year` and the GDP variable to numeric.

```
GDP_per_capita <- gdp.per.capita.worldbank[-1, ]

colnames(GDP_per_capita) <- c(
  "Entity",
  "Code",
  "Year",
  "GDP per capita, PPP (constant 2017 international $)"
)
```

```r
GDP_per_capita <- GDP_per_capita %>%
  mutate(Year = as.numeric(Year)) %>%
  mutate(`GDP per capita, PPP (constant 2017 international $)` =
           as.numeric(`GDP per capita, PPP (constant 2017 international $)`))
```

Here we combine the continent information with the GDP dataset, producing a single panel with GDP per capita, country, continent and year.

```r
GDP_with_Continents <- full_join(Sorted_continents, GDP_per_capita)
```

```
## Joining with ‘by = join_by(Entity, Code)‘
```

This chunk creates six continent-specific GDP data frames, each ordered by country and then by year. They will be used to compute growth within each region.

```r
Africa_GDP <- GDP_with_Continents %>%
  filter(Continent == "Africa") %>%
  arrange(Entity, Year)

Asia_GDP <- GDP_with_Continents %>%
  filter(Continent == "Asia") %>%
  arrange(Entity, Year)

Europe_GDP <- GDP_with_Continents %>%
  filter(Continent == "Europe") %>%
  arrange(Entity, Year)

North_America_GDP <- GDP_with_Continents %>%
  filter(Continent == "North America") %>%
  arrange(Entity, Year)

Oceania_GDP <- GDP_with_Continents %>%
  filter(Continent == "Oceania") %>%
  arrange(Entity, Year)

South_America_GDP <- GDP_with_Continents %>%
  filter(Continent == "South America") %>%
  arrange(Entity, Year)
```

We define a helper function `Growth()` that, within each country, computes GDP per capita growth (difference) and growth rate (percentage) from one year to the next. We then apply it to each continent.

```r
Growth <- function(df) {
  df %>%
    group_by(Entity) %>%
    mutate(
      GDP_Growth = `GDP per capita, PPP (constant 2017 international $)` -
        lag(`GDP per capita, PPP (constant 2017 international $)`),
      GDP_Growth_Rate =
        (`GDP per capita, PPP (constant 2017 international $)` -
           lag(`GDP per capita, PPP (constant 2017 international $)`)) /
        lag(`GDP per capita, PPP (constant 2017 international $)`) * 100
    ) %>%
    ungroup()
}

Africa_GDP_Growth_Rate        <- Growth(Africa_GDP)
Asia_GDP_Growth_Rate          <- Growth(Asia_GDP)
Europe_GDP_Growth_Rate        <- Growth(Europe_GDP)
North_America_GDP_Growth_Rate <- Growth(North_America_GDP)
Oceania_GDP_Growth_Rate       <- Growth(Oceania_GDP)
South_America_GDP_Growth_Rate <- Growth(South_America_GDP)
```

Next, we define `Average_Growth()` which collapses a continent-level data frame to average GDP growth rates per year (across its countries). We then combine all continents into one dataset.

```r
Average_Growth <- function(df) {
  df %>%
    select(Year, GDP_Growth_Rate, Continent) %>%
    group_by(Year, Continent) %>%
    summarise(Average_GDP_Growth_Rate = mean(GDP_Growth_Rate, na.rm = TRUE))
}

Continents_Growth_Rate <- rbind(
  Average_Growth(Africa_GDP_Growth_Rate),
  Average_Growth(Asia_GDP_Growth_Rate),
  Average_Growth(Europe_GDP_Growth_Rate),
  Average_Growth(North_America_GDP_Growth_Rate),
  Average_Growth(Oceania_GDP_Growth_Rate),
  Average_Growth(South_America_GDP_Growth_Rate)
)
```

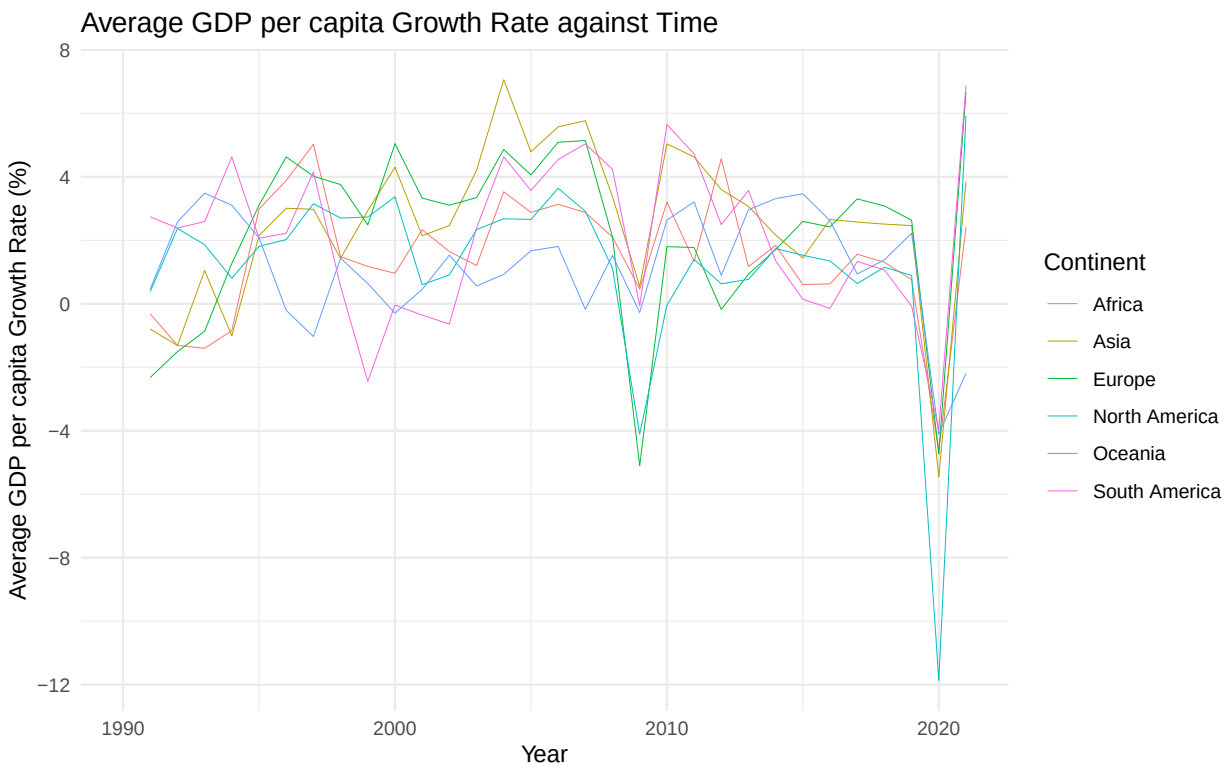
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

This graph shows the average GDP per capita growth rate over time for each continent using annual lines.

```
Graph_Continents_Growth_Rate <- ggplot(
  Continents_Growth_Rate,
  aes(x = Year, y = Average_GDP_Growth_Rate, colour = Continent)
) +
  geom_line(linewidth = 0.2) +
  labs(
    x = "Year",
    y = "Average GDP per capita Growth Rate (%)",
    title = "Average GDP per capita Growth Rate against Time"
  ) +
  theme_minimal()

Graph_Continents_Growth_Rate
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

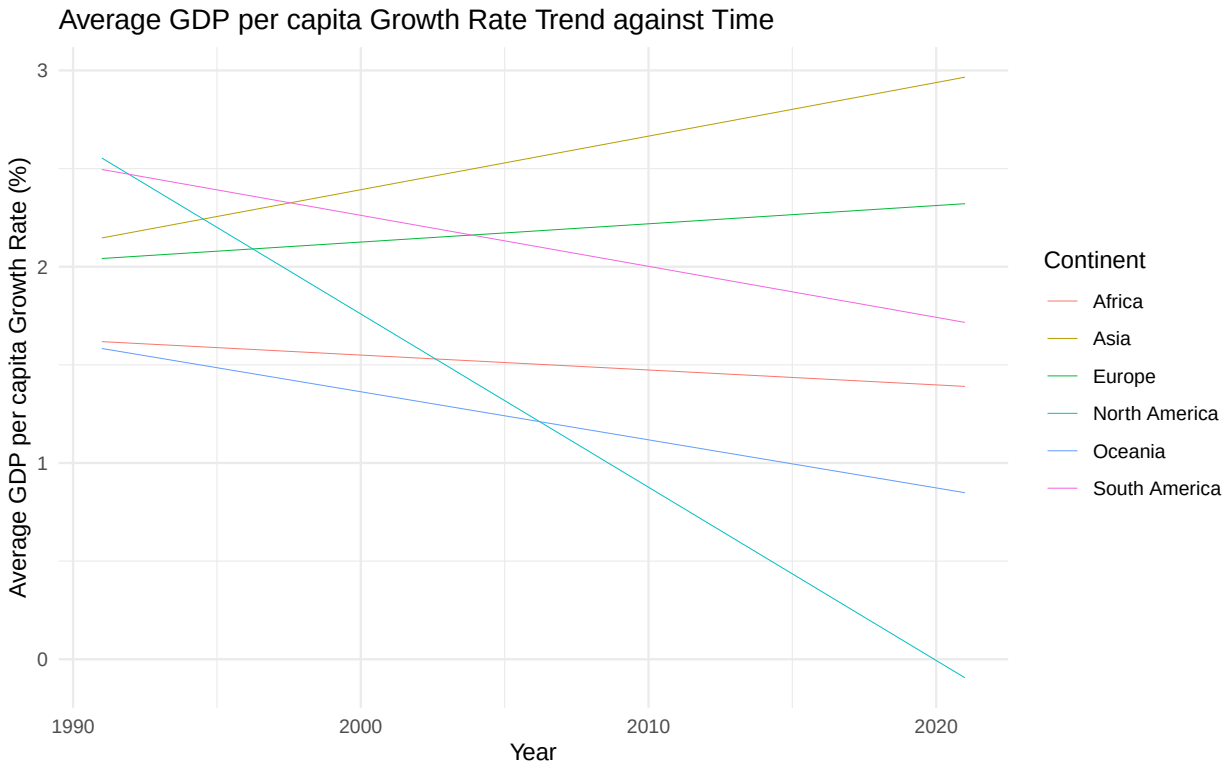


This alternative graph replaces annual lines with linear trends (fitted via OLS) to better compare the long-run growth trajectories of each continent.

```
Graph_Continents_Average_Growth_Rate <- ggplot(
  Continents_Growth_Rate,
  aes(x = Year, y = Average_GDP_Growth_Rate, colour = Continent)
) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 0.2) +
  labs(
    x = "Year",
    y = "Average GDP per capita Growth Rate (%)",
    title = "Average GDP per capita Growth Rate Trend against Time"
  ) +
  theme_minimal()

Graph_Continents_Average_Growth_Rate
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



We now define plotting functions for country-level GDP growth and its trend within each continent. These will be used to generate detailed graphs by country.

```r
Graph_Growth <- function(df, Continent) {
  ggplot(df, aes(x = Year, y = GDP_Growth, colour = Entity)) +
    geom_line(linewidth = 0.2) +
    labs(
      x = "Year",
      y = "GDP per capita Growth (PPP)",
      title = paste0("GDP per capita Growth against Time (", Continent, ")")
    ) +
    theme_minimal()
}

Graph_Average_Growth <- function(df, Continent) {
  ggplot(df, aes(x = Year, y = GDP_Growth, colour = Entity)) +
    geom_smooth(method = "lm", se = FALSE, linewidth = 0.2) +
    labs(
      x = "Year",
      y = "GDP per capita Growth (PPP)",
      title = paste0("GDP per capita Growth Trend against Time (", Continent, ")")
    ) +
    theme_minimal()
}
```

Here we actually call the functions above to produce country-level GDP growth and trend graphs for each continent (optional to print or save individually).

```r
Graph_Growth(Africa_GDP_Growth_Rate,         Continent = "Africa")
Graph_Growth(Asia_GDP_Growth_Rate,           Continent = "Asia")
Graph_Growth(Europe_GDP_Growth_Rate,         Continent = "Europe")
Graph_Growth(North_America_GDP_Growth_Rate, Continent = "North America")
Graph_Growth(Oceania_GDP_Growth_Rate,        Continent = "Oceania")
Graph_Growth(South_America_GDP_Growth_Rate, Continent = "South America")

Graph_Average_Growth(Africa_GDP_Growth_Rate,         Continent = "Africa")
Graph_Average_Growth(Asia_GDP_Growth_Rate,           Continent = "Asia")
Graph_Average_Growth(Europe_GDP_Growth_Rate,         Continent = "Europe")
Graph_Average_Growth(North_America_GDP_Growth_Rate, Continent = "North America")
Graph_Average_Growth(Oceania_GDP_Growth_Rate,        Continent = "Oceania")
Graph_Average_Growth(South_America_GDP_Growth_Rate, Continent = "South America")
```

Similarly, we define functions to plot GDP growth **rates** (rather than levels) for each country and to show smoothed trends.

```r
Graph_Growth_Rate <- function(df, Continent) {
  ggplot(df, aes(x = Year, y = GDP_Growth_Rate, colour = Entity)) +
    geom_line(linewidth = 0.2) +
    labs(
      x = "Year",
      y = "GDP per capita Growth Rate (%)",
```

```
      title = paste0("GDP per capita Growth Rate against Time (", Continent, ")")
    ) +
    theme_minimal()
}

Graph_Average_Growth_Rate <- function(df, Continent) {
  ggplot(df, aes(x = Year, y = GDP_Growth_Rate, colour = Entity)) +
    geom_smooth(method = "lm", se = FALSE, linewidth = 0.2) +
    labs(
      x = "Year",
      y = "GDP per capita Growth Rate (%)",
      title = paste0("GDP per capita Growth Rate Trend against Time (", Continent, ")")
    ) +
    theme_minimal()
}
```

This chunk illustrates how to generate those country-level growth-rate plots for each continent (again, optional).

```
Graph_Growth_Rate(Africa_GDP_Growth_Rate,        Continent = "Africa")
Graph_Growth_Rate(Asia_GDP_Growth_Rate,          Continent = "Asia")
Graph_Growth_Rate(Europe_GDP_Growth_Rate,        Continent = "Europe")
Graph_Growth_Rate(North_America_GDP_Growth_Rate, Continent = "North America")
Graph_Growth_Rate(Oceania_GDP_Growth_Rate,       Continent = "Oceania")
Graph_Growth_Rate(South_America_GDP_Growth_Rate, Continent = "South America")

Graph_Average_Growth_Rate(Africa_GDP_Growth_Rate,        Continent = "Africa")
Graph_Average_Growth_Rate(Asia_GDP_Growth_Rate,          Continent = "Asia")
Graph_Average_Growth_Rate(Europe_GDP_Growth_Rate,        Continent = "Europe")
Graph_Average_Growth_Rate(North_America_GDP_Growth_Rate, Continent = "North America")
Graph_Average_Growth_Rate(Oceania_GDP_Growth_Rate,       Continent = "Oceania")
Graph_Average_Growth_Rate(South_America_GDP_Growth_Rate, Continent = "South America")
```

Part 1 – Income groups and growth targets

We first combine all continent-specific data into one large panel of country-year observations with GDP growth and its rate.

```
Countries_GDP_Growth_Rate <- rbind(
  Africa_GDP_Growth_Rate,
  Asia_GDP_Growth_Rate,
  Europe_GDP_Growth_Rate,
  North_America_GDP_Growth_Rate,
  Oceania_GDP_Growth_Rate,
  South_America_GDP_Growth_Rate
)
```

Next we define income groups using GDP per capita thresholds, restrict to the most recent 5-year window (2017–2021), and ensure each country is assigned a single income group based on its most recent value.

```
Countries_Income_Group <- Countries_GDP_Growth_Rate %>%
  filter(between(Year, 2017, 2021)) %>%
  mutate(Income_Group = case_when(
    `GDP per capita, PPP (constant 2017 international $)` <= 1135   ~ "Low Income",
    `GDP per capita, PPP (constant 2017 international $)` <= 4495   ~ "Lower Middle Income",
    `GDP per capita, PPP (constant 2017 international $)` <= 13935  ~ "Upper Middle Income",
    `GDP per capita, PPP (constant 2017 international $)` > 13935   ~ "High Income"
  )) %>%
  group_by(Entity) %>%
  mutate(Income_Group = Income_Group[which.max(Year)]) %>%
  ungroup()
```

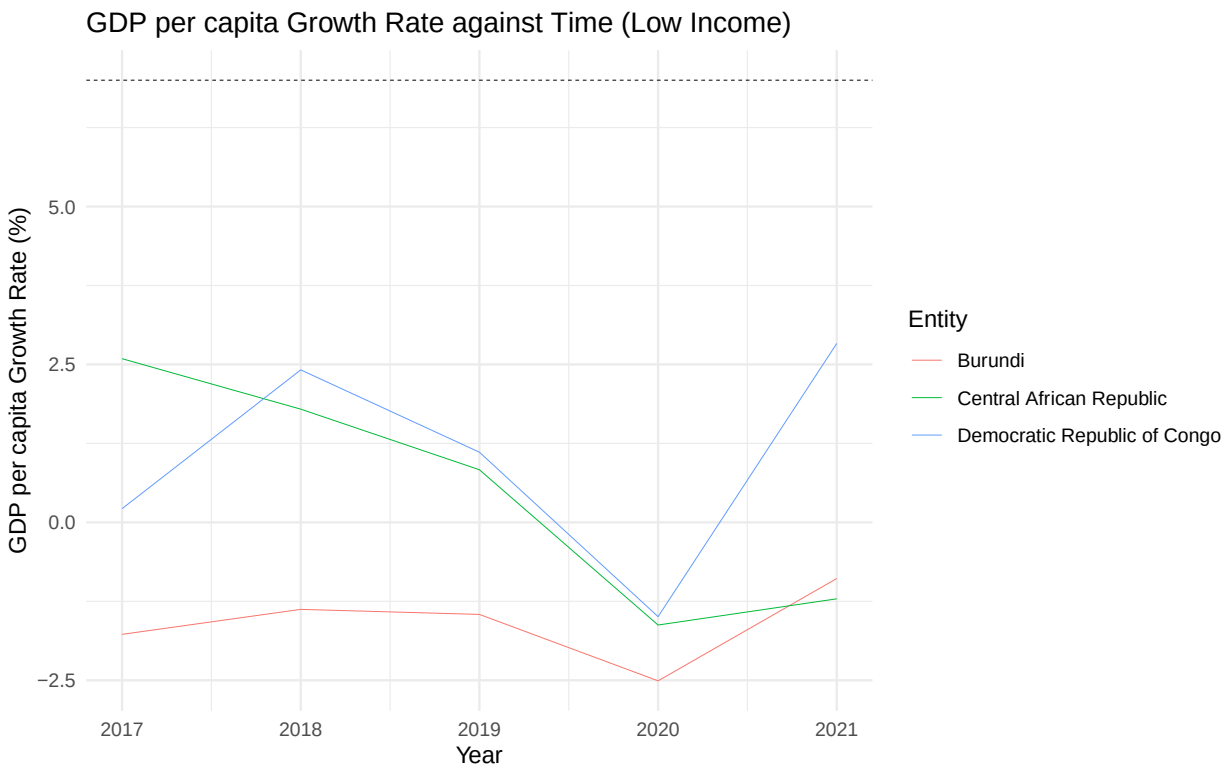
We then create four separate datasets, one for each income group. These will be used both for individual country plots and for continent-level averages within each income category.

```
Low_Income <- Countries_Income_Group %>%
  filter(Income_Group == "Low Income")

Lower_Middle_Income <- Countries_Income_Group %>%
  filter(Income_Group == "Lower Middle Income")

Upper_Middle_Income <- Countries_Income_Group %>%
  filter(Income_Group == "Upper Middle Income")

High_Income <- Countries_Income_Group %>%
  filter(Income_Group == "High Income")
```

These four graphs show GDP per capita growth rates over time for each income group, with a horizontal dashed line indicating the corresponding SDG growth target.

```
Graph_Low_Income <- ggplot(Low_Income) +
  aes(x = Year, y = GDP_Growth_Rate, colour = Entity) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 7, linetype = "dashed", colour = "black", linewidth = 0.2) +
  labs(
    x = "Year",
    y = "GDP per capita Growth Rate (%)",
    title = "GDP per capita Growth Rate against Time (Low Income)"
  ) +
  theme_minimal()

Graph_Lower_Middle_Income <- ggplot(Lower_Middle_Income) +
  aes(x = Year, y = GDP_Growth_Rate, colour = Entity) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 5, linetype = "dashed", colour = "black", linewidth = 0.2) +
```

```
  labs(
    x = "Year",
    y = "GDP per capita Growth Rate (%)",
    title = "GDP per capita Growth Rate against Time (Lower Middle Income)"
  ) +
  theme_minimal()

Graph_Upper_Middle_Income <- ggplot(Upper_Middle_Income) +
  aes(x = Year, y = GDP_Growth_Rate, colour = Entity) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 4, linetype = "dashed", colour = "black", linewidth = 0.2) +
  labs(
    x = "Year",
    y = "GDP per capita Growth Rate (%)",
    title = "GDP per capita Growth Rate against Time (Upper Middle Income)"
  ) +
  theme_minimal()

Graph_High_Income <- ggplot(High_Income) +
  aes(x = Year, y = GDP_Growth_Rate, colour = Entity) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 2, linetype = "dashed", colour = "black", linewidth = 0.2) +
  labs(
    x = "Year",
    y = "GDP per capita Growth Rate (%)",
    title = "GDP per capita Growth Rate against Time (High Income)"
  ) +
  theme_minimal()

Graph_Low_Income
```



(The other three graphs can be printed the same way if needed.)

We now compute mean GDP growth rates by continent and year within each income group, to see how regions perform relative to the target.

```
Average_Low_Income <- Low_Income %>%
  group_by(Continent, Year) %>%
  summarise(Mean_GDP_Growth_Rate = mean(GDP_Growth_Rate, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Continent'. You can override using the
## `.groups` argument.
```

```
Average_Lower_Middle_Income <- Lower_Middle_Income %>%
  group_by(Continent, Year) %>%
  summarise(Mean_GDP_Growth_Rate = mean(GDP_Growth_Rate, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Continent'. You can override using the
## `.groups` argument.
```

```
Average_Upper_Middle_Income <- Upper_Middle_Income %>%
  group_by(Continent, Year) %>%
  summarise(Mean_GDP_Growth_Rate = mean(GDP_Growth_Rate, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Continent'. You can override using the
## `.groups` argument.
```

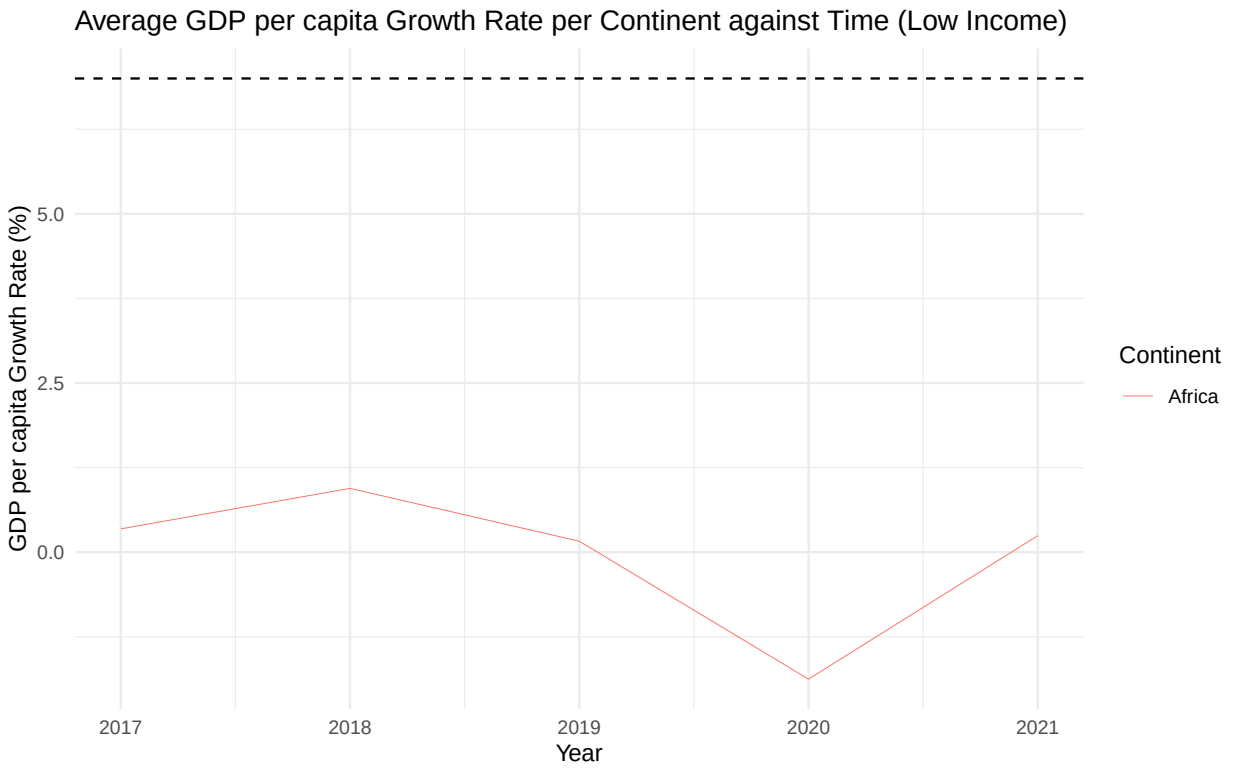
```
Average_High_Income <- High_Income %>%
  group_by(Continent, Year) %>%
  summarise(Mean_GDP_Growth_Rate = mean(GDP_Growth_Rate, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Continent'. You can override using the
## `.groups` argument.
```

These four graphs show the continent-level mean growth rate for each income group, again with horizontal dashed target lines.

```
Graph_Average_Low_Income <- ggplot(Average_Low_Income) +
  aes(x = Year, y = Mean_GDP_Growth_Rate, colour = Continent) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 7, linetype = "dashed", colour = "black") +
  labs(
    title = "Average GDP per capita Growth Rate per Continent against Time (Low Income)",
    x = "Year",
    y = "GDP per capita Growth Rate (%)"
  ) +
  theme_minimal()

Graph_Average_Lower_Middle_Income <- ggplot(Average_Lower_Middle_Income) +
  aes(x = Year, y = Mean_GDP_Growth_Rate, colour = Continent) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 5, linetype = "dashed", colour = "black") +
  labs(
    title = "Average GDP per capita Growth Rate per Continent against Time (Lower Middle Income)",
    x = "Year",
    y = "GDP per capita Growth Rate (%)"
  ) +
  theme_minimal()

Graph_Average_Upper_Middle_Income <- ggplot(Average_Upper_Middle_Income) +
  aes(x = Year, y = Mean_GDP_Growth_Rate, colour = Continent) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 4, linetype = "dashed", colour = "black") +
  labs(
    title = "Average GDP per capita Growth Rate per Continent against Time (Upper Middle Income)",
    x = "Year",
    y = "GDP per capita Growth Rate (%)"
  ) +
  theme_minimal()

Graph_Average_High_Income <- ggplot(Average_High_Income) +
  aes(x = Year, y = Mean_GDP_Growth_Rate, colour = Continent) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 2, linetype = "dashed", colour = "black") +
```

```
  labs(
    title = "Average GDP per capita Growth Rate per Continent against Time (High Income)",
    x = "Year",
    y = "GDP per capita Growth Rate (%)"
  ) +
  theme_minimal()

Graph_Average_Low_Income
```



This chunk counts how many countries each continent has in total. We will use this as the denominator when computing proportions that meet their growth targets.

```
Total_countries_in_Continent <- Sorted_continents %>%
  group_by(Continent) %>%
  summarise(Number_Countries = n())
```

Here we calculate, for each income group, how many countries per continent meet their respective growth target in each year (7%, 5%, 4%, or 2%).

```
Low_Income_With_Growth <- Low_Income %>%
  filter(GDP_Growth_Rate >= 7) %>%
  group_by(Year, Continent) %>%
  summarise(Number = length(unique(Entity))) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
Lower_Middle_Income_With_Growth <- Lower_Middle_Income %>%
  filter(GDP_Growth_Rate >= 5) %>%
  group_by(Year, Continent) %>%
  summarise(Number = length(unique(Entity))) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
Upper_Middle_Income_With_Growth <- Upper_Middle_Income %>%
  filter(GDP_Growth_Rate >= 4) %>%
  group_by(Year, Continent) %>%
  summarise(Number = length(unique(Entity))) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
High_Income_With_Growth <- High_Income %>%
  filter(GDP_Growth_Rate >= 2) %>%
  group_by(Year, Continent) %>%
  summarise(Number = length(unique(Entity))) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

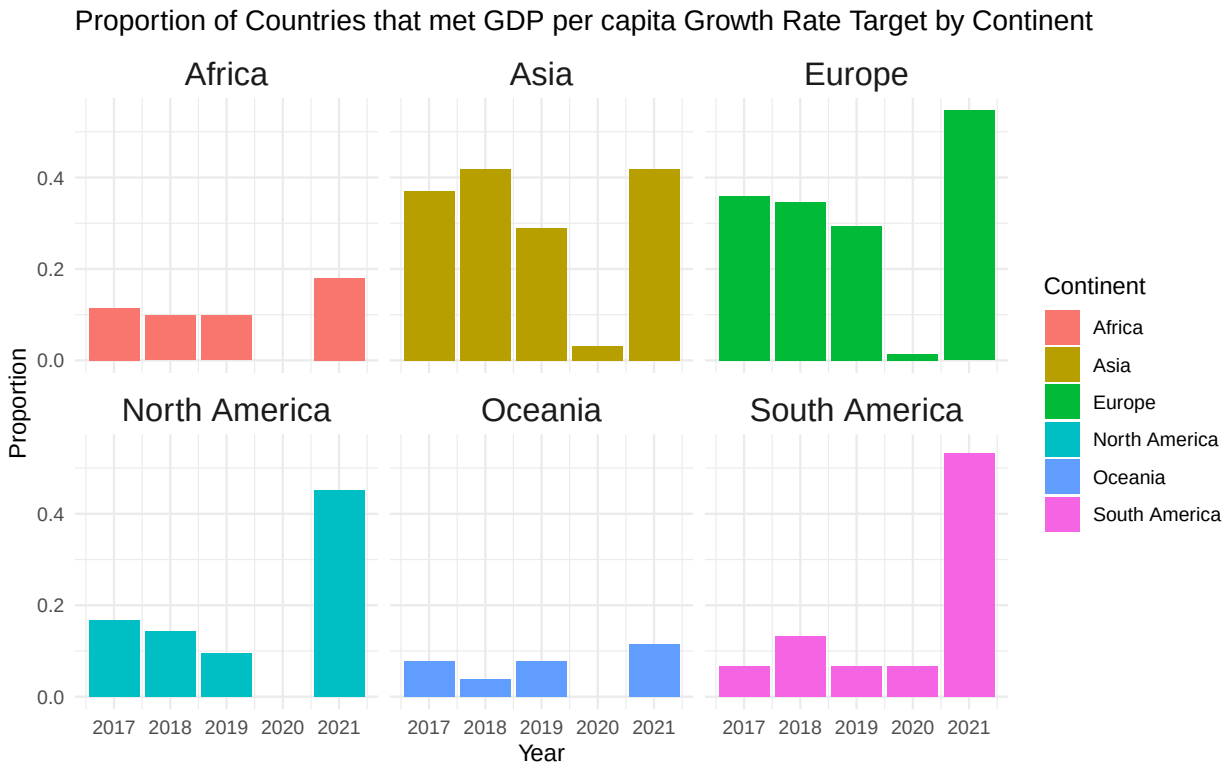
```
Achieved_Growth <- rbind(
  Low_Income_With_Growth,
  Lower_Middle_Income_With_Growth,
  Upper_Middle_Income_With_Growth,
  High_Income_With_Growth
)
```

We now convert these counts into proportions of countries per continent that meet the GDP growth target each year, and we visualise them with faceted bar charts.

```
Proportion_Achieved_Growth <- Achieved_Growth %>%
  group_by(Year, Continent) %>%
  summarise(Total_Number = sum(Number)) %>%
  ungroup() %>%
  left_join(Total_countries_in_Continent, by = "Continent") %>%
  mutate(Proportion = Total_Number / Number_Countries)
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```r
Graph_Proportion_Achieved_Growth <- ggplot(Proportion_Achieved_Growth) +
  aes(x = Year, y = Proportion, fill = Continent) +
  geom_col() +
  facet_wrap(~Continent) +
  labs(
    x = "Year",
    y = "Proportion",
    title = "Proportion of Countries that met GDP per capita Growth Rate Target by Continent"
  ) +
  theme_minimal() +
  theme(strip.text = element_text(size = 15))

Graph_Proportion_Achieved_Growth
```



Part 2 – Least Developed Countries (LDCs)

We import the UNSD methodology file that lists which countries are LDCs, clean the table, and keep only the LDC rows.

```r
Additional_csv_1 <- read.csv("~/Documents/GitHub/Data-Science-Coursework/UNSD - Methodology.csv",
                             header = FALSE, sep = ";")

colnames(Additional_csv_1) <- Additional_csv_1[1, ]
```

```
Additional_csv_1$"Least Developed Countries (LDC)" <-
  as.character(Additional_csv_1$"Least Developed Countries (LDC)")

Additional_csv_1$"Least Developed Countries (LDC)"[
  Additional_csv_1$"Least Developed Countries (LDC)" == ""
] <- NA

LDCs <- Additional_csv_1[-1, ] %>%
  select("Country or Area", "Least Developed Countries (LDC)") %>%
  filter(!is.na(`Least Developed Countries (LDC)`))
```

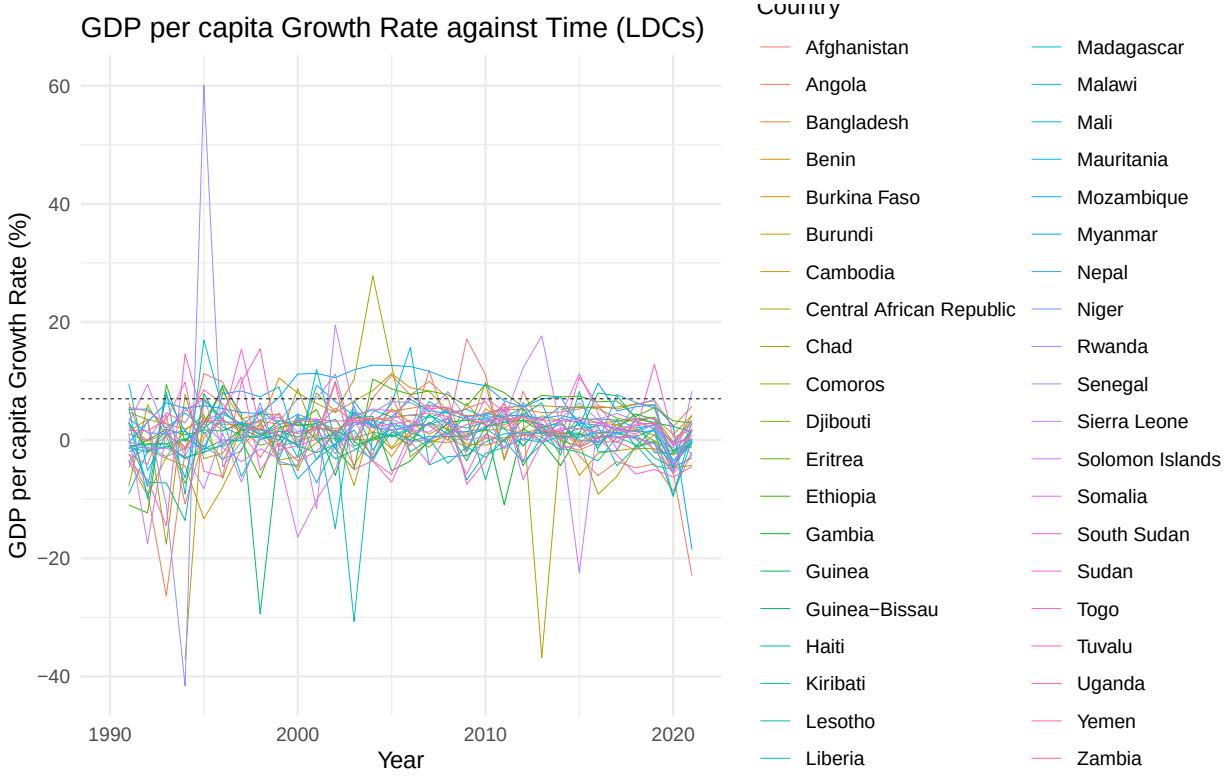
Here we merge the LDC list with the full GDP growth dataset, obtaining GDP growth rates only for LDCs.

```
LDCs_GDP_Growth_Rate <- inner_join(
  LDCs,
  Countries_GDP_Growth_Rate,
  by = c("Country or Area" = "Entity")
)

LDCs_GDP_Growth_Rate <- LDCs_GDP_Growth_Rate[, -2]
```

This plot shows GDP per capita growth rates over time for all LDCs, with a dashed 7% target line.

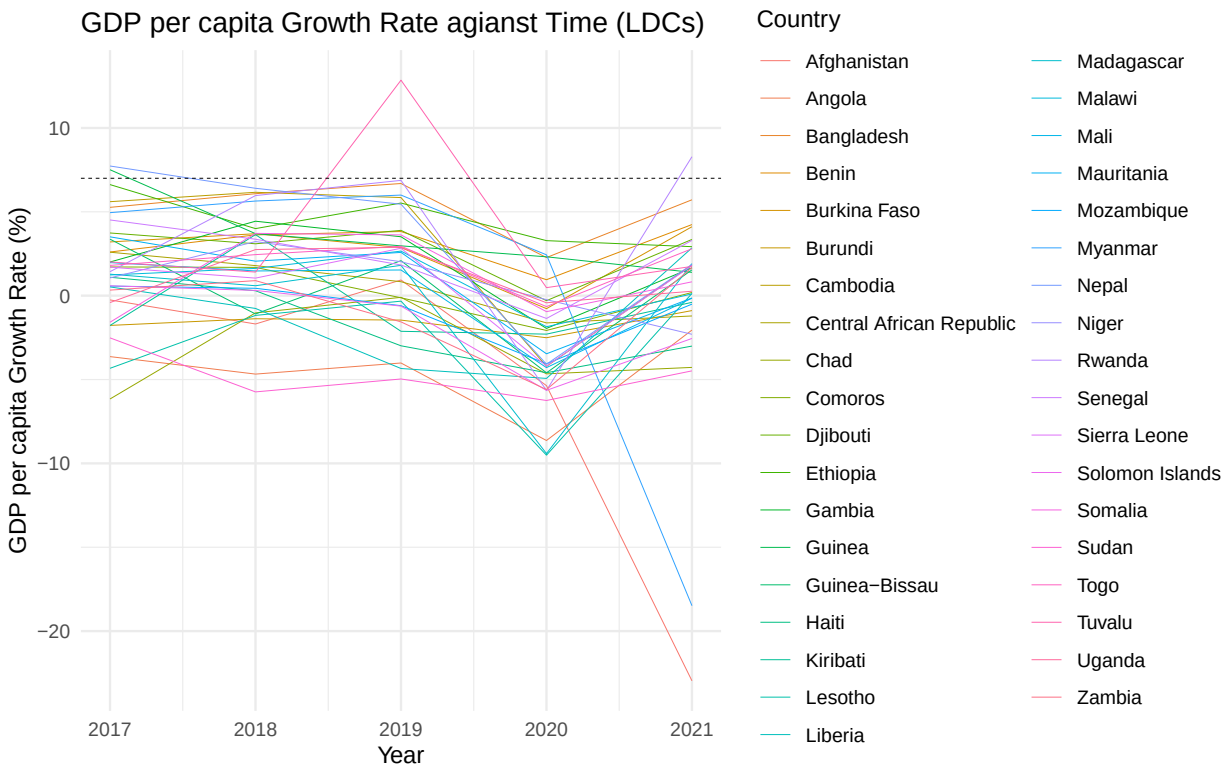
```
Graph_LDCs <- ggplot(LDCs_GDP_Growth_Rate) +
  aes(x = Year, y = GDP_Growth_Rate,
      group = `Country or Area`, colour = `Country or Area`) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 7, linetype = "dashed",
             colour = "black", linewidth = 0.2) +
  labs(
    title = "GDP per capita Growth Rate against Time (LDCs)",
    x = "Year",
    y = "GDP per capita Growth Rate (%)",
    colour = "Country"
  ) +
  theme_minimal()

Graph_LDCs
```

```
## Warning: Removed 40 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



We then restrict attention to the most recent 5 years for LDCs, based on the latest year available in the data, and plot the recent-period growth rates.

```
max_year <- max(LDCs_GDP_Growth_Rate$Year, na.rm = TRUE)

LDC_GDP_Growth_Recent <- LDCs_GDP_Growth_Rate %>%
  filter(Year >= max_year - 4)

Graph_LDCs_Recent <- ggplot(LDC_GDP_Growth_Recent) +
  aes(x = Year, y = GDP_Growth_Rate,
      group = `Country or Area`, colour = `Country or Area`) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 7, linetype = "dashed",
             colour = "black", linewidth = 0.2) +
  labs(
    title = "GDP per capita Growth Rate agianst Time (LDCs)",
    x = "Year",
    y = "GDP per capita Growth Rate (%)",
    colour = "Country"
  ) +
  theme_minimal()

Graph_LDCs_Recent
```



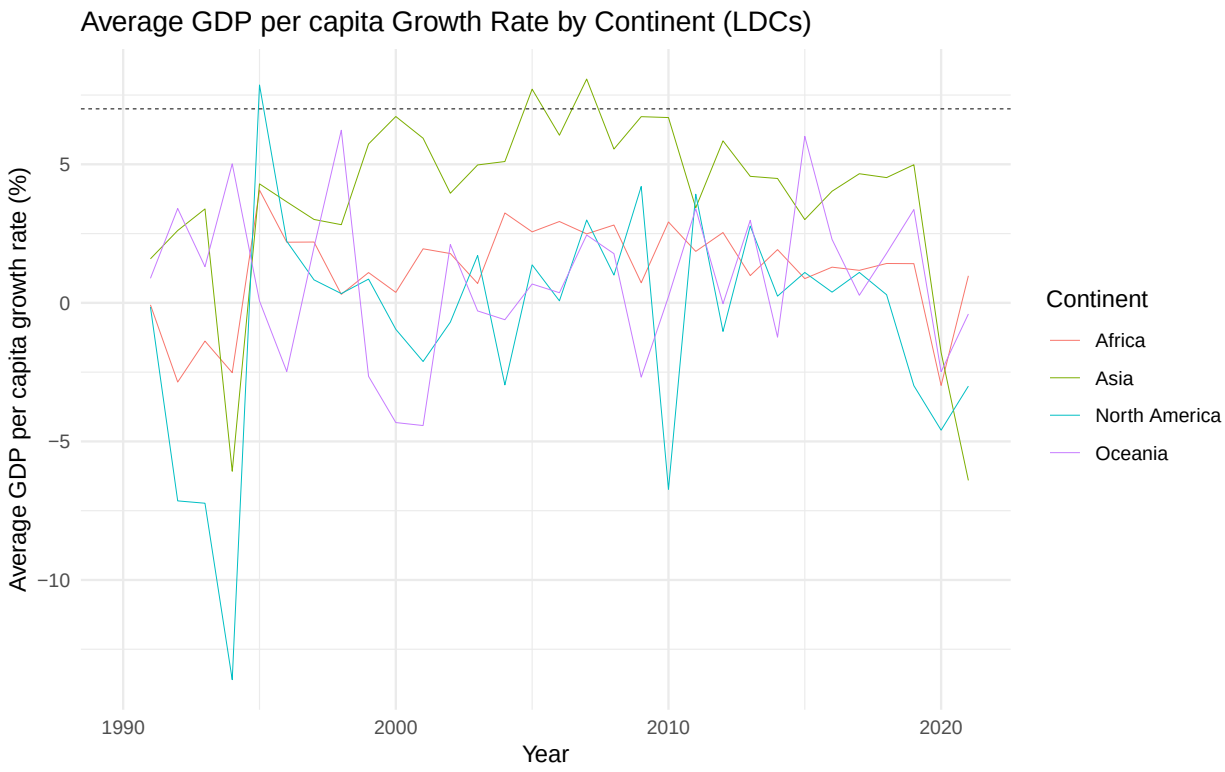
Here we compute average GDP growth rates by continent among LDCs, and plot how these averages evolve over time with the 7% target line.

```
Continents_LDC_Growth_Rate <- LDCs_GDP_Growth_Rate %>%
  group_by(Continent, Year) %>%
  summarise(Mean_Growth = mean(GDP_Growth_Rate, na.rm = TRUE)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Continent'. You can override using the
## `.groups` argument.
```

```
Graph_Continents_LDC_Growth_Rate <- ggplot(Continents_LDC_Growth_Rate) +
  aes(x = Year, y = Mean_Growth, colour = Continent) +
  geom_line(linewidth = 0.2) +
  geom_hline(yintercept = 7, linetype = "dashed",
             colour = "black", linewidth = 0.2) +
  labs(
    title = "Average GDP per capita Growth Rate by Continent (LDCs)",
    x = "Year",
    y = "Average GDP per capita growth rate (%)",
    colour = "Continent"
  ) +
  theme_minimal()

Graph_Continents_LDC_Growth_Rate
```

```
## Warning: Removed 6 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



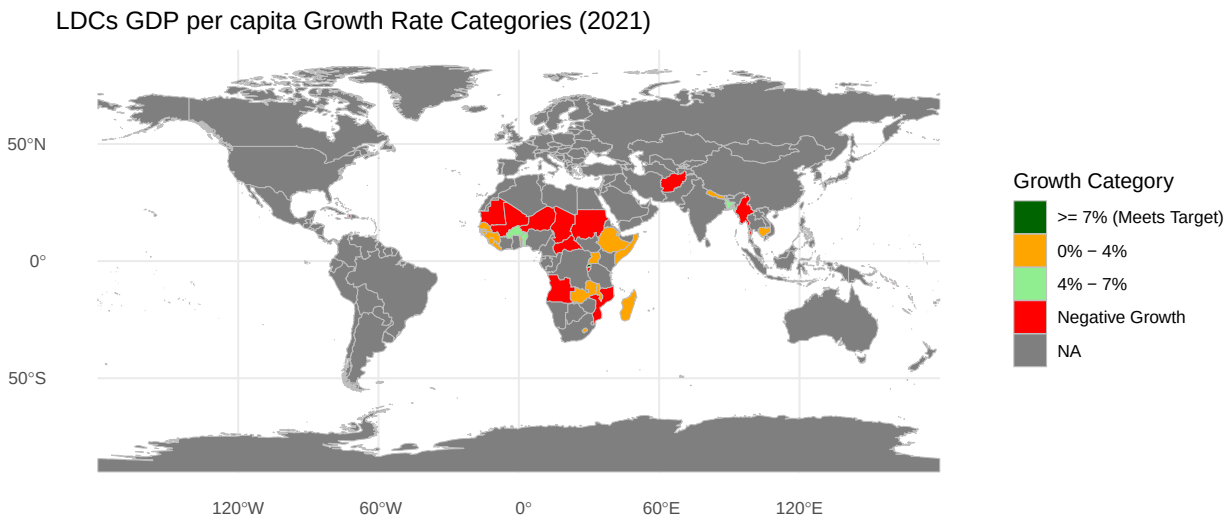
Next we keep only year 2021 for LDCs and classify them into growth-rate categories, which will be used for a world map.

```
LDCs_GDP_Growth_Rate_2021 <- LDCs_GDP_Growth_Rate %>%
  filter(Year == 2021)

LDCs_GDP_Growth_Rate_2021 <- LDCs_GDP_Growth_Rate_2021 %>%
  mutate(Growth_Category = case_when(
    GDP_Growth_Rate >= 7  ~ ">= 7% (Meets Target)",
    GDP_Growth_Rate >= 4  ~ "4% - 7%",
    GDP_Growth_Rate >= 0  ~ "0% - 4%",
    TRUE                  ~ "Negative Growth"
  ))
```

This chunk constructs a world map using `rnaturalearth`, joins it with the LDC growth categories, and visualises the distribution of LDC growth performance in 2021.

```
World <- ne_countries(scale = "medium", returnclass = "sf")

World_LDCs <- World %>%
```

```
  left_join(LDCs_GDP_Growth_Rate_2021, by = c("iso_a3" = "Code"))

Graph_World <- ggplot(World_LDCs) +
  geom_sf(aes(fill = Growth_Category), color = "grey", size = 0.1) +
  scale_fill_manual(
    values = c(
      ">= 7% (Meets Target)" = "darkgreen",
      "4% - 7%"              = "lightgreen",
      "0% - 4%"              = "orange",
      "Negative Growth"      = "red"
    )
  ) +
  labs(
    title = "LDCs GDP per capita Growth Rate Categories (2021)",
    fill  = "Growth Category"
  ) +
  theme_minimal()

Graph_World
```



Finally, we look at how many LDCs **achieve or fail** the 7% target over 2017–2021 by continent. We first classify each observation by "Meets target" vs "Below target", then count the number of LDCs in each category.

```
LDCs_According_Growth <- LDC_GDP_Growth_Recent %>%
  mutate(
    Target = ifelse(GDP_Growth_Rate >= 7,
                    "Meets target (>= 7%)",
```

```
                    "Below target")
  ) %>%
  group_by(Year, Target, Continent) %>%
  summarise(LDCs_Number = length(unique(Code))) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year', 'Target'. You can override using
## the `.groups` argument.
```

We construct a full grid of all combinations of year, target status and continent (to include zero counts), build a custom combined `Year_Status` variable, and prepare labels for a two-row x-axis.

```
All_Years      <- 2017:2021
All_Targets    <- c("Meets target (>= 7%)", "Below target")
All_Continents <- unique(LDC_GDP_Growth_Recent$Continent)

Full_Grid <- expand.grid(
  Year      = All_Years,
  Target    = All_Targets,
  Continent = All_Continents,
  stringsAsFactors = FALSE
)

LDCs_According_Growth_Complete <- Full_Grid %>%
  left_join(LDCs_According_Growth,
            by = c("Year", "Target", "Continent")) %>%
  mutate(LDCs_Number = ifelse(is.na(LDCs_Number), 0, LDCs_Number))

LDCs_According_Growth_Complete <- LDCs_According_Growth_Complete %>%
  mutate(
    Growth_Status = ifelse(Target == "Meets target (>= 7%)", "Achieved", "Did Not Achieve"),
    Year_Status   = paste0(Year, " (", Growth_Status, ")")
  )

Order <- LDCs_According_Growth_Complete %>%
  arrange(Year, desc(Growth_Status)) %>%
  pull(Year_Status) %>%
  unique()

LDCs_According_Growth_Complete$Year_Status <- factor(
  LDCs_According_Growth_Complete$Year_Status,
  levels = Order
)

x_labels <- LDCs_According_Growth_Complete %>%
  distinct(Year_Status, Year, Growth_Status)
```

This final graph displays, for each year, how many LDCs achieved or did not achieve the 7% growth target, broken down by continent. The x-axis uses two rows of labels: year and achievement status.

```r
Graph_LDCs_Growth_Status <- ggplot(LDCs_According_Growth_Complete) +
  aes(x = Year_Status, y = LDCs_Number, fill = Continent) +
  geom_bar(stat = "identity") +
  labs(
    x = "Year and Achievement Status",
    y = "Number of LDCs",
    title = "Number of LDCs that did not achieve or achieved 7% GDP per capita Growth Rate",
    fill = "Continent"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    plot.margin = margin(20, 20, 40, 20)
  ) +
  geom_text(aes(x = Year_Status, y = -0.5, label = Year),
            data = x_labels, inherit.aes = FALSE, vjust = 1.2) +
  geom_text(aes(x = Year_Status, y = -1.5, label = Growth_Status),
            data = x_labels, inherit.aes = FALSE, vjust = 1.2) +
  coord_cartesian(clip = "off")

Graph_LDCs_Growth_Status
```