INTRODUCTION TO DATA SCIENCE

Measuring progress towards the UN Sustainable Development Goal 8: Decent work and economic growth

Group K10

10 Dec 2025

This is an R Markdown files containing the code used to produce all of the graphs in the report.

Instructions for use

1. Download the repository and unzip inside of an R project
2. Set the working directory to the folder named after the repository
3. Run `finalScript.R` which has the code shown below
4. Type `figure_#` into the console where '#' is the number of the figure you want to access (3 - 11, 12a, 12b)

Output Files

1. `continent_mean_yoy_neet_pct_change_consecutive`: Summary statistics of year-on-year NEET change by continent
2. `continent_population_weighted_neet_change`: Summary statistics of NEET levels by continent
3. `countries_closest_to_10_percent`: NEET data of the 5 closets countries to 10% NEET reduction
4. `countries_closest_to_30_percent`: NEET data of the 5 closets countries to 30% NEET reduction
5. `countries_near_10_or_30_threshold`: Combination of the previous two files
6. `countries_with_consecutive_year_data`: Counts the number of consecutive years for which there is NEET data per country
7. `data_summary.csv`: Summary statistics of the master data set
8. `global_mean_yoy_neet_pct_change_consecutive`: Summary statistics of the number of consecutive years for which there is NEET data per country
9. `global_yoy_pct_by_year_consecutive`: Summary statistics of the number of consecutive years for which there is NEET data per year, globally
10. `master_dataset.csv`: Master data set containing every data point from every original data set
11. `ldc_status_by_country.csv`: Describes the continent and LDC status of every country

finalScript.R:

```
#
======================================================================
======
# Introduction to Data Science - Group Assignment
# Data Cleaning and Merging

# Install required packages
# install.packages('tidyverse')
# install.packages('dplyr')
# install.packages('readr')
# install.packages('ggplot2')
# install.packages('scales')
# install.packages('maps')
# install.packages('ggrepel')

# Load required packages
library(tidyverse)
library(dplyr)
library(readr)
library(ggplot2)
library(scales)
library(maps)
library(ggrepel)

#
======================================================================
======
# PART 1: Load Data
#
======================================================================
======

cat("=== STEP 1: LOADING DATA ===\n\n")

# Load the three required datasets
gdp <- read_csv("gdp-per-capita-worldbank.csv")
neet <- read_csv("youth-not-in-education-employment-training.csv")
continent <- read_csv("continents-according-to-our-world-in-data.csv")

# Load additional datasets
edu_raw <- read_csv("government_expenditure_on_education.csv", skip =
4)
ldc_data <- read_csv("ldc_data.csv")
sdi_data <- read_csv("sustainable_development_index.csv")
population_raw <- read_csv("population_data.csv")

cat("GDP data:", nrow(gdp), "rows\n")
cat("NEET data:", nrow(neet), "rows\n")
cat("Continent data:", nrow(continent), "rows\n")
cat("Education data:", nrow(edu_raw), "rows\n")
cat("LDC classification:", nrow(ldc_data), "rows\n")
cat("SDI data:", nrow(sdi_data), "rows\n")
cat("Population data:", nrow(population_raw), "rows\n\n")

#
======================================================================
======
# PART 2: Data Cleaning and Renaming
#
======================================================================
======

cat("=== STEP 2: CLEANING DATA ===\n\n")

# Clean GDP data
gdp_clean <- gdp %>%
  rename(
    country = Entity,
    code = Code,
    year = Year,
    gdp_per_capita = `GDP per capita, PPP (constant 2017 international
$)`
  ) %>%
  distinct(code, year, .keep_all = TRUE)

# Clean NEET data
neet_clean <- neet %>%
  rename(
    country = Entity,
    code = Code,
    year = Year,
    neet_share = `Share of youth not in education, employment or
training, total (% of youth population)`
  ) %>%
  distinct(code, year, .keep_all = TRUE)

# Clean continent classification data
continent_clean <- continent %>%
  rename(
    country = Entity,
    code = Code,
    year = Year,
    continent = Continent
  ) %>%
  filter(continent != "Antarctica") %>%
  select(code, continent) %>%
  distinct()
```

```
# Clean education expenditure data
edu_long <- edu_raw %>%
  select(`Country Name`, `Country Code`, matches("^\\d{4}$")) %>%
  pivot_longer(
    cols = -c(`Country Name`, `Country Code`),
    names_to = "year",
    values_to = "edu_expenditure_gdp"
  ) %>%
  rename(country = `Country Name`, code = `Country Code`) %>%
  mutate(year = as.integer(year)) %>%
  distinct(code, year, .keep_all = TRUE)

# Clean LDC classification data
ldc_clean <- ldc_data %>%
  rename(
    ccode = CCODE,
    iso3 = `ISO -3`,
    country_ldc = Countries,
    ldc_status = Status
  ) %>%
  mutate(
    code = iso3,
    ldc_status = toupper(trimws(ldc_status)),
    is_ldc = ldc_status == "LDC"
  ) %>%
  select(code, ldc_status, is_ldc) %>%
  distinct()

cat("LDC classification cleaned\n")
cat("  - LDC countries:", sum(ldc_clean$is_ldc), "\n")
cat("  - ODC/Other countries:", sum(!ldc_clean$is_ldc), "\n\n")

# Clean SDI data
sdi_clean <- sdi_data %>%
  rename(code = iso) %>%
  pivot_longer(
    cols = matches("^\\d{4}$"),
    names_to = "year",
    values_to = "sdi_score"
  ) %>%
  mutate(
    year = as.integer(year),
    sdi_score = as.numeric(sdi_score)
  ) %>%
  filter(!is.na(sdi_score), year >= 2000) %>%
  select(code, year, sdi_score) %>%
  distinct(code, year, .keep_all = TRUE)

cat("SDI data cleaned\n")
cat("  - Year range:", min(sdi_clean$year, na.rm = TRUE), "to",
    max(sdi_clean$year, na.rm = TRUE), "\n")
cat("  - Countries:", n_distinct(sdi_clean$code), "\n\n")

#
======================================================================
======
# Clean Population data - WORLD BANK WIDE FORMAT
#
======================================================================
======

cat("Cleaning population data (World Bank format)...\n")

# Read with skip to handle metadata rows
pop_raw <- read_csv("population_data.csv", skip = 4, show_col_types =
FALSE)

cat("Original data:", nrow(pop_raw), "rows,", ncol(pop_raw), "columns\
n")

# World Bank format typically has:
# Country Name | Country Code | Indicator Name | Indicator Code | 1960
| 1961 | ...

# Find the country code column (usually 2nd column)
code_col_idx <- 2

# Find year columns (4-digit numbers)
all_cols <- colnames(pop_raw)
year_cols <- all_cols[grepl("^[12][0-9]{3}$", all_cols)]

if(length(year_cols) == 0) {
  cat("Warning: No year columns found with 4-digit pattern.\n")
  cat("Trying to detect year columns by position (columns 5
onwards)...\n")
  # Assume years start from column 5
  year_cols <- colnames(pop_raw)[5:ncol(pop_raw)]
  # Filter to only numeric columns
  year_cols <- year_cols[sapply(pop_raw[year_cols], is.numeric)]
}

cat("Found", length(year_cols), "year columns\n")
cat("Year range:", head(year_cols, 1), "to", tail(year_cols, 1), "\n")

# Convert to long format
population_clean <- pop_raw %>%
  # Select country code + all year columns
  select(code = all_of(code_col_idx), all_of(year_cols)) %>%
  # Convert to long format
  pivot_longer(
    cols = all_of(year_cols),
    names_to = "year",
    values_to = "population"
  ) %>%
  # Clean data types
  mutate(
    code = as.character(code),
    year = as.integer(year),
    population = as.numeric(population)
  ) %>%
  # Remove missing values
  filter(!is.na(code), !is.na(year), !is.na(population)) %>%
  # Remove duplicates
  distinct(code, year, .keep_all = TRUE)

# Verify success
if(nrow(population_clean) == 0) {
  cat("\n□ ERROR: Cleaning produced 0 rows!\n")
  cat("Sample of original data:\n")
  print(head(pop_raw[, 1:5], 3))
  stop("Population cleaning failed")
}

year_min <- min(population_clean$year, na.rm = TRUE)
year_max <- max(population_clean$year, na.rm = TRUE)

if(is.infinite(year_min)) {
  cat("\n□ ERROR: Year conversion failed\n")
  stop("Could not extract years from column names")
}

cat("✓ Population data cleaned\n")
cat("  Year range:", year_min, "to", year_max, "\n")
cat("  Countries:", n_distinct(population_clean$code), "\n")
cat("  Total observations:", nrow(population_clean), "\n\n")

#
======================================================================
======
# PART 3: Merge All Datasets
#
======================================================================
======

cat("=== STEP 3: MERGING ALL DATASETS ===\n\n")

# Merge step by step with clear documentation
master_data <- gdp_clean %>%
  # Step 1: Merge NEET data
  left_join(
    neet_clean,
    by = c("code", "year"),
    suffix = c("_gdp", "_neet")
  ) %>%
  # Step 2: Merge education expenditure data
  left_join(
    edu_long,
    by = c("code", "year")
  ) %>%
  # Step 3: Merge LDC classification (by code only)
  left_join(
    ldc_clean,
    by = "code"
  ) %>%
  # Step 4: Merge SDI data
  left_join(
    sdi_clean,
    by = c("code", "year")
  ) %>%
  # Step 5: Merge Population data (NEW!)
  left_join(
    population_clean,
    by = c("code", "year")
  ) %>%
  # Step 6: Merge continent classification
  left_join(
    continent_clean,
    by = "code"
  ) %>%
  # Handle duplicate country columns and set default values
  mutate(
    country = coalesce(country_gdp, country_neet, country),
    ldc_status = ifelse(is.na(ldc_status), "UNKNOWN", ldc_status),
    is_ldc = ifelse(is.na(is_ldc), FALSE, is_ldc)
  ) %>%
  # Reorganize columns in logical order
  select(
    # Identifiers
    country, code, year, continent,
    # Development status
    ldc_status, is_ldc,
    # Economic indicators
    gdp_per_capita,
    # Population (NEW!)
    population,
    # Employment/Education indicators
    neet_share, edu_expenditure_gdp,
    # Sustainability indicator
```

```
    sdi_score
  ) %>%
  # Keep observations with at least one piece of data
  filter(!is.na(gdp_per_capita) | !is.na(neet_share) |
           !is.na(edu_expenditure_gdp) | !is.na(sdi_score) | !
is.na(population)) %>%
  # Keep only data with continent classification
  filter(!is.na(continent))

cat("Master dataset created successfully!\n")
cat("Total observations:", nrow(master_data), "\n")
cat("Countries:", n_distinct(master_data$code), "\n")
cat("Year range:", min(master_data$year), "to", max(master_data$year),
"\n\n")

#
======================================================================
======
# PART 4: Data Quality Check
#
======================================================================
======

cat("=== STEP 4: DATA QUALITY CHECK ===\n\n")

# View structure
glimpse(master_data)

# Check LDC distribution
cat("\n--- LDC Status Distribution ---\n")
ldc_summary <- master_data %>%
  distinct(code, .keep_all = TRUE) %>%
  count(ldc_status) %>%
  mutate(percentage = round(n / sum(n) * 100, 1))
print(ldc_summary)

# Check data completeness by continent
cat("\n--- Data Completeness by Continent ---\n")
data_coverage <- master_data %>%
  group_by(continent) %>%
  reframe(
    n_countries = n_distinct(code),
    n_ldc = sum(is_ldc, na.rm = TRUE),
    n_observations = n(),
    gdp_coverage = round(sum(!is.na(gdp_per_capita)) / n() * 100, 1),
    neet_coverage = round(sum(!is.na(neet_share)) / n() * 100, 1),
    edu_coverage = round(sum(!is.na(edu_expenditure_gdp)) / n() * 100,
1),
    sdi_coverage = round(sum(!is.na(sdi_score)) / n() * 100, 1),
    pop_coverage = round(sum(!is.na(population)) / n() * 100, 1),  #
```

```
NEW!
    .groups = "drop"
  )
print(data_coverage)

# Check missing values
cat("\n--- Missing Value Statistics ---\n")
master_data %>%
  reframe(
    total_obs = n(),
    gdp_missing = sum(is.na(gdp_per_capita)),
    neet_missing = sum(is.na(neet_share)),
    edu_missing = sum(is.na(edu_expenditure_gdp)),
    sdi_missing = sum(is.na(sdi_score)),
    pop_missing = sum(is.na(population)),  # NEW!
    ldc_unknown = sum(ldc_status == "UNKNOWN")
  ) %>%
  print()

# Check population coverage for LDCs (important for weighted analysis)
cat("\n--- Population Coverage for LDCs ---\n")
ldc_pop_check <- master_data %>%
  filter(is_ldc == TRUE, year >= 2015, year <= 2023) %>%
  reframe(
    total_obs = n(),
    pop_available = sum(!is.na(population)),
    pop_coverage_pct = round(pop_available / total_obs * 100, 1)
  )
print(ldc_pop_check)

#
======================================================================
======
# PART 5: Save Cleaned Data
#
======================================================================
======

cat("\n=== STEP 5: SAVING CLEANED DATA ===\n\n")

# Save master dataset
write_csv(master_data, "output/master_dataset.csv")

# Save a summary for reference
summary_stats <- data.frame(
  Dataset = c("Total Observations", "Unique Countries", "Year Range",
              "LDC Countries", "Non-LDC Countries", "Continents",
              "Population Coverage (%)"),
  Value = c(
    nrow(master_data),
    n_distinct(master_data$code),
    paste(min(master_data$year), "-", max(master_data$year)),
    sum(master_data %>% distinct(code, .keep_all = TRUE) %>%
pull(is_ldc)),
    sum(!(master_data %>% distinct(code, .keep_all = TRUE) %>%
pull(is_ldc))),
    n_distinct(master_data$continent),
    round(sum(!is.na(master_data$population)) / nrow(master_data) *
100, 1)
  )
)

write_csv(summary_stats, "output/data_summary.csv")
cat("✓ Saved: data_summary.csv\n\n")

# Load data
master_data <- read_csv("output/master_dataset.csv", show_col_types =
FALSE)
master <- master_data

# 4. GDP per capita growth rate (%) over time by continent
cont_master <- master %>% group_by(continent, year) %>%
  mutate(gdp_per_capita = mean(gdp_per_capita, na.rm = T),
    neet_share = mean(neet_share, na.rm = T),
    du_expenditure_gdp = mean(edu_expenditure_gdp, na.rm = T),
  ) %>%
  ungroup() %>% select(year:edu_expenditure_gdp) %>%
  distinct(year, continent, .keep_all = T)
cont_master <- cont_master %>%
  group_by(continent) %>%
  arrange(year) %>%
  mutate(gdp_growth_rate = (gdp_per_capita - lag(gdp_per_capita)) /
lag(gdp_per_capita) * 100) %>%
  ungroup()

# 5. NEET change 2010 → 2020 (bar chart)
neet_change <- master %>%
  filter(year %in% c(2010, 2020)) %>%
  group_by(continent, year) %>%
  reframe(avg_neet = mean(neet_share, na.rm = T), .groups = 'drop') %>
%
  pivot_wider(names_from = year, values_from = avg_neet, names_prefix
= "year_") %>%
  mutate(neet_change = year_2020 - year_2010) %>%
  filter(!is.na(neet_change))

#11: Calculate GDP growth rate for each country, then average by
continent
gdp_growth_data <- master %>%
  group_by(country) %>%
  arrange(year) %>%
  mutate(gdp_growth_rate = (gdp_per_capita - lag(gdp_per_capita)) /
lag(gdp_per_capita) * 100) %>%
  ungroup() %>%
  group_by(continent, year) %>%
  reframe(avg_growth_rate = mean(gdp_growth_rate, na.rm =
TRUE), .groups = 'drop')

gdp_growth_data <- gdp_growth_data %>%
  group_by(continent) %>%
  arrange(year) %>%
  mutate(growth_5yr_avg = zoo::rollmean(avg_growth_rate, k = 5, fill =
NA, align = "center")) %>%
  ungroup()

#
======================================================================
======
# DATA ANALYSIS FOR TARGET 1 REPORT
# Quick checks and summary statistics
#
======================================================================
======

# Calculate growth rates
analysis_data <- master_data %>%
  filter(!is.na(gdp_per_capita)) %>%
  arrange(code, year) %>%
  group_by(code) %>%
  mutate(gdp_growth_rate = (gdp_per_capita / lag(gdp_per_capita) - 1)
* 100) %>%
  ungroup()

# Focus on 2015-2023
analysis_period <- analysis_data %>%
  filter(year >= 2015, year <= 2023, !is.na(gdp_growth_rate))

#
======================================================================
======
# SUMMARY STATS
#
======================================================================
======

cat("\n=== BASIC STATS ===\n")
cat("Period:", min(analysis_period$year), "-",
max(analysis_period$year), "\n")
cat("Total countries:", n_distinct(analysis_period$code), "\n")
cat("LDCs:", sum(analysis_period$is_ldc), "observations\n\n")
```

```
#
======================================================================
======
# TARGET ACHIEVEMENT RATES
#
======================================================================
======

cat("=== TARGET ACHIEVEMENT (7% threshold) ===\n\n")

# By continent
achievement_by_continent <- analysis_period %>%
  filter(is_ldc == TRUE) %>%
  group_by(continent) %>%
  reframe(
    n = n(),
    above_7 = sum(gdp_growth_rate >= 7, na.rm = TRUE),
    pct = round(above_7/n * 100, 1)
  ) %>%
  arrange(desc(pct))

print(achievement_by_continent)

# Overall
overall <- analysis_period %>%
  filter(is_ldc == TRUE) %>%
  reframe(
    total = n(),
    above_7 = sum(gdp_growth_rate >= 7, na.rm = TRUE),
    pct = round(above_7/total * 100, 1)
  )

cat("\nOverall LDCs:", overall$pct, "% of observations ≥7%\n")

#
======================================================================
======
# COUNTRY AVERAGES
#
======================================================================
======

cat("\n=== AVERAGE GROWTH BY COUNTRY (2015-2023) ===\n\n")

country_avg <- analysis_period %>%
  filter(is_ldc == TRUE) %>%
  group_by(country, code) %>%
  reframe(
    avg = round(mean(gdp_growth_rate, na.rm = TRUE), 2),
    n = n(),
    .groups = "drop"
  ) %>%
  filter(n >= 3)

# How many above 7%?
above_7_countries <- country_avg %>% filter(avg >= 7)
cat("LDCs with average ≥7%:", nrow(above_7_countries), "\n")
if(nrow(above_7_countries) > 0) print(above_7_countries)

# Top 5
cat("\nTop 5:\n")
top5 <- country_avg %>% arrange(desc(avg)) %>% head(5)
print(top5)

# Bottom 5
cat("\nBottom 5:\n")
bottom5 <- country_avg %>% arrange(avg) %>% head(5)
print(bottom5)

#
======================================================================
======
# COVID IMPACT
#
======================================================================
======

cat("\n=== COVID IMPACT ===\n\n")

covid_impact <- analysis_period %>%
  filter(is_ldc == TRUE) %>%
  mutate(period = case_when(
    year <= 2019 ~ "Pre-COVID",
    year == 2020 ~ "COVID-2020",
    year >= 2021 ~ "Post-COVID"
  )) %>%
  group_by(continent, period) %>%
  reframe(avg = round(mean(gdp_growth_rate, na.rm = TRUE), 1), .groups
= "drop")

# Asia
cat("Asia:\n")
print(covid_impact %>% filter(continent == "Asia"))

# Africa
cat("\nAfrica:\n")
print(covid_impact %>% filter(continent == "Africa"))
```

```
# Check 2020
cat("\n2020 by continent:\n")
covid_2020 <- analysis_period %>%
  filter(is_ldc == TRUE, year == 2020) %>%
  group_by(continent) %>%
  reframe(avg_2020 = round(mean(gdp_growth_rate, na.rm = TRUE), 1))
print(covid_2020)

#
======================================================================
======
# VARIATION ANALYSIS
#
======================================================================
======

cat("\n=== WITHIN VS BETWEEN CONTINENT VARIATION ===\n\n")

# Add continent to country averages
country_avg_cont <- country_avg %>%
  left_join(
    analysis_period %>% filter(is_ldc == TRUE) %>%
      select(code, continent) %>% distinct(),
    by = "code"
  )

# Within-continent SD
within_sd <- country_avg_cont %>%
  filter(!is.na(continent)) %>%
  group_by(continent) %>%
  reframe(sd = round(sd(avg, na.rm = TRUE), 2))

cat("Within-continent SD:\n")
print(within_sd)
cat("Average:", round(mean(within_sd$sd, na.rm = TRUE), 2), "\n")

# Between-continent SD
continent_means <- country_avg_cont %>%
  filter(!is.na(continent)) %>%
  group_by(continent) %>%
  reframe(mean = mean(avg, na.rm = TRUE))

between_sd <- sd(continent_means$mean, na.rm = TRUE)
cat("\nBetween-continent SD:", round(between_sd, 2), "\n")

#
======================================================================
======
# SDI ANALYSIS
#
======================================================================
======

cat("\n=== SDI RANGE ===\n\n")

sdi_stats <- analysis_period %>%
  filter(is_ldc == TRUE, !is.na(sdi_score)) %>%
  group_by(continent) %>%
  reframe(
    min = round(min(sdi_score), 2),
    q25 = round(quantile(sdi_score, 0.25), 2),
    median = round(median(sdi_score), 2),
    q75 = round(quantile(sdi_score, 0.75), 2),
    max = round(max(sdi_score), 2),
    n = n()
  )

print(sdi_stats)

#
======================================================================
======
# NON-LDC COMPARISON
#
======================================================================
======

cat("\n=== NON-LDC GROWTH (Europe & N.America) ===\n\n")

non_ldc <- analysis_period %>%
  filter(is_ldc == FALSE, continent %in% c("Europe", "North America"))
%>%
  reframe(
    median = round(median(gdp_growth_rate, na.rm = TRUE), 1),
    mean = round(mean(gdp_growth_rate, na.rm = TRUE), 1),
    sd = round(sd(gdp_growth_rate, na.rm = TRUE), 1),
    q25 = round(quantile(gdp_growth_rate, 0.25, na.rm = TRUE), 1),
    q75 = round(quantile(gdp_growth_rate, 0.75, na.rm = TRUE), 1)
  )

print(non_ldc)

cat("\n=== DONE ===\n")

#
======================================================================
======
# TARGET 1: FIGURES FOR REPORT
# UN SDG 8.1 - GDP Growth in Least Developed Countries
#
======================================================================
======

cat("\n")
cat("=================================================================
===========\n")
cat("              TARGET 1: CREATING FIGURES 3-6 FOR REPORT
\n")
cat("=================================================================
===========\n\n")

#
======================================================================
======
# STEP 1: LOAD DATA
#
======================================================================
======

cat("✓ Data loaded successfully\n")
cat("  Total observations:", nrow(master_data), "\n")
cat("  Countries:", n_distinct(master_data$code), "\n")
cat("  Year range:", min(master_data$year), "to",
max(master_data$year), "\n\n")

#
======================================================================
======
# STEP 2: PREPARE DATA FOR ANALYSIS
#
======================================================================
======

cat("Step 2: Calculating GDP growth rates...\n")

# Calculate GDP growth rates
analysis_data <- master_data %>%
  filter(!is.na(gdp_per_capita)) %>%
  arrange(code, year) %>%
  group_by(code) %>%
  mutate(gdp_growth_rate = (gdp_per_capita / lag(gdp_per_capita) - 1)
* 100) %>%
  ungroup()

# Auto-detect available GDP data year range
gdp_year_range <- analysis_data %>%
  filter(!is.na(gdp_per_capita)) %>%
  reframe(
    min_year = min(year, na.rm = TRUE),
    max_year = max(year, na.rm = TRUE)
  )

cat("✓ GDP data year range detected:", gdp_year_range$min_year, "to",
gdp_year_range$max_year, "\n")

# Focus on SDG period: 2015 onwards, up to max available year
analysis_period <- analysis_data %>%
  filter(year >= 2015, year <= gdp_year_range$max_year, !
is.na(gdp_growth_rate))

# Create year label for figures
year_label <- paste0("2015-", gdp_year_range$max_year)

cat("✓ Data prepared\n")
cat("  Analysis period:", year_label, "\n")
cat("  Observations with growth data:", nrow(analysis_period), "\n\n")

#
======================================================================
======
# STEP 3: CREATE FIGURES
#
======================================================================
======

cat("Step 3: Creating figures...\n\n")

#
----------------------------------------------------------------------
------
# FIGURE 3: GDP GROWTH TRENDS BY CONTINENT (was Figure 1)
# Population-weighted for Non-LDCs, simple average for LDCs
#
----------------------------------------------------------------------
------
cat("Creating Figure 3: GDP Growth Trends by Continent...\n")

growth_trends <- analysis_period %>%
  filter(!is.na(continent), !is.na(is_ldc), !is.na(population)) %>%
  group_by(continent, is_ldc, year) %>%
  reframe(
    # LDCs: Simple average (each country = 1 unit)
    # Non-LDCs: Population-weighted (larger countries have more
weight)
    avg_growth = ifelse(
      first(is_ldc),
      mean(gdp_growth_rate, na.rm = TRUE),
      sum(gdp_growth_rate * population, na.rm = TRUE) /
        sum(population, na.rm = TRUE)
    ),
    .groups = "drop"
  ) %>%
  mutate(group_label = ifelse(is_ldc, "LDCs", "Non-LDCs"))

# figure 3
graph1 <- ggplot(growth_trends, aes(x = year, y = avg_growth, color =
group_label)) +
  # COVID-19 period highlight (2019-2020)
  annotate("rect",
           xmin = 2019, xmax = 2020,
           ymin = -Inf, ymax = Inf,
           fill = "gray", alpha = 0.15) +

  # COVID label
  annotate("text",
           x = 2019.5, y = Inf,
           label = "COVID-19",
           color = "gray30",
           size = 4,
           vjust = 1.5,
           fontface = "italic") +

  # Main lines (SOLID)
  geom_line(linewidth = 1.4) +
  geom_point(size = 3) +

  # Reference lines: 7% target (red dashed) and 2% baseline (blue
dashed)
  geom_hline(yintercept = 7, linetype = "dashed", color = "#e74c3c",
linewidth = 1.2) +
  geom_hline(yintercept = 2, linetype = "dashed", color = "#3498db",
linewidth = 1.2) +

  # Labels for reference lines
  annotate("text", x = 2015.5, y = 7.5, label = "7% Target",
           color = "#e74c3c", size = 4.5, hjust = 0, fontface =
"bold") +
  annotate("text", x = 2015.5, y = 2.5, label = "2% Baseline",
           color = "#3498db", size = 4, hjust = 0, fontface = "bold")
+

  facet_wrap(~continent, ncol = 3) +

  labs(
    title = "Figure 3: GDP Per Capita Growth Rates by Continent",
    subtitle = paste0("LDCs vs Non-LDCs (", year_label, ") | LDCs: 
simple avg; Non-LDCs: population-weighted"),
    x = "Year",
    y = "Average GDP Growth Rate (%)",
    color = "Country Group"
  ) +

  scale_color_manual(values = c("LDCs" = "#e74c3c", "Non-LDCs" = 
"#3498db")) +

  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 18),
    plot.subtitle = element_text(size = 12),
    legend.position = "bottom",
    legend.title = element_text(size = 14, face = "bold"),
    legend.text = element_text(size = 13),
    strip.text = element_text(face = "bold", size = 14),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 13),
    axis.text.x = element_text(size = 12)
  )

# ggsave("output/figure3_growth_trends.png", graph1, width = 14, 
height = 10, dpi = 300)
cat("  ✓ Saved: figure3_growth_trends.png\n")
cat("    Year range:", year_label, "\n")
cat("    Method: LDCs (simple avg), Non-LDCs (population-weighted)\n\
n")

# 
-----------------------------------------------------------------------
------
# FIGURE 4: GROWTH DISTRIBUTION BY CONTINENT (was Figure 2)
# 
-----------------------------------------------------------------------
------
growth_box <- analysis_period %>%
  filter(!is.na(gdp_growth_rate), !is.na(continent), !is.na(is_ldc)) 
%>%
  mutate(group_label = ifelse(is_ldc, "LDCs", "Non-LDCs"))

# figure 4
graph2 <- ggplot(growth_box, aes(x = continent, y = gdp_growth_rate, 
fill = group_label)) +
  geom_boxplot(outlier.alpha = 0.3, width = 0.6,
               position = position_dodge(width = 0.7)) +

  # Reference lines: 7% target (red dashed) and 2% baseline (blue 
dashed)
```

```
  geom_hline(yintercept = 7, linetype = "dashed", color = "#e74c3c",
linewidth = 1.2) +
  geom_hline(yintercept = 2, linetype = "dashed", color = "#3498db",
linewidth = 1.2) +

  labs(
    title = "Figure 4: Distribution of GDP Growth by Continent",
    subtitle = paste0("LDCs vs Non-LDCs (", year_label, ") | Red
dashed: 7% target; Blue dashed: 2% baseline"),
    x = "Continent",
    y = "GDP Growth Rate (%)",
    fill = "Country Group"
  ) +
  scale_fill_manual(values = c("LDCs" = "#e74c3c", "Non-LDCs" =
"#3498db")) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 18),
    plot.subtitle = element_text(size = 12),
    legend.position = "bottom",
    legend.title = element_text(size = 14, face = "bold"),
    legend.text = element_text(size = 13),
    axis.title.x = element_text(size = 15, face = "bold"),
    axis.title.y = element_text(size = 15, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 13),
    axis.text.y = element_text(size = 13),
    strip.text = element_text(face = "bold")
  ) +
  coord_cartesian(ylim = c(-10, 10))

# ggsave("output/figure4_growth_distribution.png", graph2, width = 10,
height = 7, dpi = 300)
cat("  ✓ Saved: figure4_growth_distribution.png\n\n")

#
----------------------------------------------------------------------
------
# FIGURE 5: SHARE OF LDCs ACHIEVING 7% TARGET (was Figure 3)
#
----------------------------------------------------------------------
------
cat("Creating Figure 5: Share of LDCs Achieving Target...\n")

ldc_count_by_continent <- analysis_period %>%
  filter(is_ldc == TRUE) %>%
  distinct(code, continent) %>%
  group_by(continent) %>%
  reframe(n_ldcs = n_distinct(code), .groups = "drop")

target_achievement <- analysis_period %>%
  filter(!is.na(continent), is_ldc == TRUE) %>%
  group_by(continent) %>%
  reframe(
    total_obs = n(),
    above_7pct = sum(gdp_growth_rate >= 7, na.rm = TRUE),
    pct_achieving = round((above_7pct / total_obs) * 100, 1),
    .groups = "drop"
  ) %>%
  left_join(ldc_count_by_continent, by = "continent")

graph3 <- ggplot(target_achievement, # figure 5
               aes(x = reorder(continent, pct_achieving),
                   y = pct_achieving,
                   fill = continent)) +
  geom_bar(stat = "identity", width = 0.7, show.legend = FALSE) +

  geom_text(aes(label = paste0(
    round(pct_achieving, 1), "%\n",
    "(", n_ldcs, " LDCs | ", total_obs, " obs)"
  )), hjust = -0.1, size = 5) +

  coord_flip() +

  labs(
    title = "Figure 5: Share of LDC Country-Years Achieving 7%
Growth",
    subtitle = paste0("By Continent (", year_label, ") | 42 of 45 UN
LDCs analyzed | Numbers: % (LDC count | obs)"),
    x = NULL,
    y = "Percentage Achieving ≥7% Growth (%)"
  ) +

  scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
  scale_fill_brewer(palette = "Set2") +

  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 18),
    plot.subtitle = element_text(size = 11),
    axis.title.x = element_text(size = 15, face = "bold"),
    axis.text.x = element_text(size = 13),
    axis.text.y = element_text(size = 14, face = "bold")
  )

# ggsave("output/figure5_target_achievement.png", graph3, width = 10,
height = 6, dpi = 300)
cat("  ✓ Saved: figure5_target_achievement.png\n\n")

#
----------------------------------------------------------------------
```

```
------
# FIGURE 6: SUSTAINABLE DEVELOPMENT vs ECONOMIC GROWTH (was Figure
4/5)
#
-----------------------------------------------------------------------
------

cat("Creating Figure 6: Sustainable Development vs Growth...\n")

sdi_growth <- analysis_period %>%
  filter(!is.na(sdi_score), !is.na(continent)) %>%
  mutate(group_label = ifelse(is_ldc, "LDCs", "Non-LDCs"))

# Calculate R values for each continent and group
r_values <- sdi_growth %>%
  group_by(continent, group_label) %>%
  summarise(
    r = cor(sdi_score, gdp_growth_rate, use = "complete.obs"),
    .groups = "drop"
  ) %>%
  mutate(
    r_label = paste0("R: ", round(r, 3))
  )

# Split into LDCs and Non-LDCs for separate layers
r_ldcs <- r_values %>%
  filter(group_label == "LDCs") %>%
  mutate(y_pos = 9.5, x_pos = 0.15)

r_non_ldcs <- r_values %>%
  filter(group_label == "Non-LDCs") %>%
  mutate(y_pos = 8.5, x_pos = 0.15)

# Recreate Figure 6 with R values
graph4 <- ggplot(sdi_growth, aes(x = sdi_score, y = gdp_growth_rate,
                              color = group_label)) +
  geom_point(alpha = 0.4, size = 2.5) +
  geom_smooth(method = "lm", se = TRUE, linewidth = 1.3) +

  # Reference lines: 7% target (red dashed) and 2% baseline (blue
dashed)
  geom_hline(yintercept = 7, linetype = "dashed", color = "#e74c3c",
linewidth = 1.2) +
  geom_hline(yintercept = 2, linetype = "dashed", color = "#3498db",
linewidth = 1.2) +

  # Add R value labels for LDCs (red)
  geom_text(data = r_ldcs,
            aes(x = x_pos, y = y_pos, label = r_label),
            color = "#e74c3c",
            hjust = 0, vjust = 0, size = 4, fontface = "bold",
            inherit.aes = FALSE) +

  # Add R value labels for Non-LDCs (blue)
  geom_text(data = r_non_ldcs,
            aes(x = x_pos, y = y_pos, label = r_label),
            color = "#3498db",
            hjust = 0, vjust = 0, size = 4, fontface = "bold",
            inherit.aes = FALSE) +

  facet_wrap(~continent, ncol = 3) +

  labs(
    title = "Figure 6: Sustainable Development vs Economic Growth",
    subtitle = paste0("SDI Score vs GDP Growth (", year_label, ") |
Red dashed: 7% target; Blue dashed: 2% baseline"),
    x = "Sustainable Development Index (Higher = More Sustainable)",
    y = "GDP Growth Rate (%)",
    color = "Country Group"
  ) +
  scale_color_manual(values = c("LDCs" = "#e74c3c", "Non-LDCs" =
"#3498db")) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 18),
    plot.subtitle = element_text(size = 11),
    legend.position = "bottom",
    legend.title = element_text(size = 14, face = "bold"),
    legend.text = element_text(size = 13),
    strip.text = element_text(face = "bold", size = 14),
    axis.title.x = element_text(size = 14, face = "bold"),
    axis.title.y = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12)
  ) +
  coord_cartesian(ylim = c(-10, 10))
# ggsave("output/figure6_sdi_vs_growth.png", graph4, width = 14,
height = 10, dpi = 300)
cat("  ✓ Saved: figure6_sdi_vs_growth.png\n\n")

# List of LDCs (as of recent UN classification)
# You'll need to check which countries in your dataset are classified
as LDCs
ldc_countries <- c(
  "Afghanistan", "Angola", "Bangladesh", "Benin", "Bhutan", "Burkina
Faso",
  "Burundi", "Cambodia", "Central African Republic", "Chad",
"Comoros",
  "Congo, Dem. Rep.", "Djibouti", "Eritrea", "Ethiopia", "Gambia",
"Guinea",
  "Guinea-Bissau", "Haiti", "Kiribati", "Laos", "Lesotho", "Liberia",
  "Madagascar", "Malawi", "Mali", "Mauritania", "Mozambique",
"Myanmar",
  "Nepal", "Niger", "Rwanda", "Sao Tome and Principe", "Senegal",
  "Sierra Leone", "Solomon Islands", "Somalia", "South Sudan",
"Sudan",
  "Tanzania", "Timor-Leste", "Togo", "Tuvalu", "Uganda", "Yemen",
"Zambia"
)

# Get unique countries from your dataset
all_countries_in_data <- master %>%
  distinct(country, continent)

# Mark which countries in your dataset are LDCs
countries_ldc_status <- all_countries_in_data %>%
  mutate(
    is_ldc = country %in% ldc_countries,
    ldc_status = ifelse(is_ldc, "LDC", "Non-LDC")
  )

print("LDCs in your dataset:")
print(countries_ldc_status %>% filter(is_ldc))

# Get world map data
world_map <- map_data("world")

# Match country names for mapping
countries_ldc_status <- countries_ldc_status %>%
  mutate(region = case_when(
    country == "United States" ~ "USA",
    country == "United Kingdom" ~ "UK",
    country == "Czechia" ~ "Czech Republic",
    country == "Congo, Dem. Rep." ~ "Democratic Republic of the
Congo",
    country == "Tanzania" ~ "Tanzania",
    country == "Laos" ~ "Laos",
    country == "Myanmar" ~ "Myanmar",
    country == "South Sudan" ~ "South Sudan",
    TRUE ~ country
  ))

# Join with map data
map_data_ldc <- world_map %>%
  left_join(countries_ldc_status, by = "region")

# Summary statistics
ldc_summary <- countries_ldc_status %>%
  group_by(continent, ldc_status) %>%
  reframe(n_countries = n(), .groups = 'drop') %>%
  pivot_wider(names_from = ldc_status, values_from = n_countries,
values_fill = 0)

print("\nLDC distribution by continent:")
print(ldc_summary)

# Save
write_csv(countries_ldc_status, "output/ldc_status_by_country.csv")

# # Step 1: Get ALL countries from master dataset (2010-2021) with
their most recent population
# all_countries_with_pop <- master %>%
#   filter(year >= 2010 & year <= 2021) %>%
#   group_by(country, continent) %>%
#   arrange(desc(year)) %>%
#   reframe(
#     most_recent_pop = first(population[!is.na(population)]),
#     .groups = 'drop'
#   ) %>%
#   filter(!is.na(most_recent_pop))  # Only keep countries with at
least some population data
#
# # Step 2: Calculate NEET change for countries WITH data
# neet_country_change <- master %>%
#   filter(!is.na(neet_share)) %>%
#   group_by(country, continent) %>%
#   arrange(year) %>%
#   reframe(
#     baseline_neet = first(neet_share[year >= 2010 & year <= 2015]),
#     baseline_year = first(year[year >= 2010 & year <= 2015]),
#     baseline_pop = first(population[year >= 2010 & year <= 2015 & !
is.na(population)]),
#     recent_neet = ifelse(
#       any(year == 2018 & !is.na(neet_share)),
#       neet_share[year == 2018][1],
#       last(neet_share[year >= 2016 & year < 2018])
#     ),
#     recent_year = ifelse(
#       any(year == 2018 & !is.na(neet_share)),
#       2018,
#       last(year[year >= 2016 & year < 2018])
#     ),
#     recent_pop = ifelse(
#       any(year == 2018 & !is.na(population)),
#       population[year == 2018][1],
#       last(population[year >= 2016 & year < 2018 & !
is.na(population)])
#     ),
#     .groups = 'drop'
#   ) %>%
#   mutate(
#     pct_change_neet = ifelse(!is.na(baseline_neet) & !
is.na(recent_neet),
#                              ((recent_neet - baseline_neet) /
baseline_neet) * 100,
#                              NA_real_),
#     target_reduction = case_when(
#       is.na(baseline_neet) ~ NA_real_,
#       baseline_neet < 10 ~ -10,
#       baseline_neet >= 10 & baseline_neet <= 30 ~ -20,
#       baseline_neet > 30 ~ -10,
#       TRUE ~ NA_real_
#     ),
#     target_status = case_when(
#       is.na(baseline_neet) | is.na(recent_neet) ~ "No Data",
#       pct_change_neet <= target_reduction ~ "Achieved",
#       TRUE ~ "Missed"
#     )
#   )
#
# # Step 3: Join ALL countries with NEET change results
# countries_complete <- all_countries_with_pop %>%
#   left_join(
#     neet_country_change %>% select(country, target_status,
recent_pop),
#     by = "country"
#   ) %>%
#   mutate(
#     target_status = ifelse(is.na(target_status), "No Data",
target_status),
#     pop_for_calculation = coalesce(recent_pop, most_recent_pop)
#   )
#
# # Step 4: Calculate total population by continent (including No Data
countries)
# continent_total_pop <- countries_complete %>%
#   group_by(continent) %>%
#   mutate(total_continent_pop = sum(pop_for_calculation, na.rm =
TRUE))
#
# # Step 5: Calculate population-weighted percentages INCLUDING "No
Data"
# population_weighted_summary_complete <- countries_complete %>%
#   left_join(continent_total_pop, by = "country") %>%
#   rename(pop_for_calculation = pop_for_calculation.x) %>%
#   group_by(continent.x, target_status.x) %>%
#   mutate(
#     total_pop_in_category = sum(pop_for_calculation, na.rm = TRUE),
#     total_continent_pop = first(total_continent_pop),
#   ) %>%
#   mutate(
#     percentage = (total_pop_in_category / total_continent_pop) * 100
#   )
#
# print(population_weighted_summary_complete)
#
# print("Population-weighted target achievement by continent
(including No Data):")
# print(population_weighted_summary_complete)
#
# # Step 6: Create stacked bar chart with all three categories
# # ggplot(population_weighted_summary_complete, aes(x = continent, y
= percentage, fill = target_status.y)) +
# #   geom_col(position = "stack", alpha = 0.8) +
# #   scale_fill_manual(
# #     values = c(
# #       "Achieved" = "#2166ac",
# #       "Missed" = "#b2182b",
# #       "No Data" = "grey80"
# #     )
# #   ) +
# #   theme_minimal() +
# #   theme(
# #     panel.grid.major.x = element_blank(),
# #     panel.grid.major.y = element_line(colour = "grey70"),
# #     panel.grid.minor = element_line(colour = "grey90", linewidth =
0.4),
# #     legend.position = "bottom",
# #     legend.direction = "horizontal",
# #     plot.title = element_text(hjust = 0.5),
# #     axis.text.x = element_text(angle = 45, hjust = 1)
# #   ) +
# #   labs(
# #     x = "Continent",
# #     y = "Percentage of Total Population (%)",
# #     fill = "Target Achievement",
# #     title = "NEET Reduction Target Achievement by Continent
(Population-Weighted)",
# #     subtitle = "Target: <10% → 10% reduction | 10-30% → 20%
reduction | >30% → 10% reduction\n(Including all countries with
population data)"
# #   ) +
# #   geom_text(
# #     aes(label = ifelse(percentage > 5, paste0(round(percentage,
1), "%"), "")),
# #     position = position_stack(vjust = 0.5),
# #     color = "white",
# #     fontface = "bold",
# #     size = 3
# #   )
# Step 1: Get ALL countries with continent info (don't filter out NAs
yet)
neet_population_weighted <- master %>%
  group_by(country, continent) %>%
  arrange(year) %>%
  reframe(
    # Baseline period (2010-2015)
    baseline_neet = last(neet_share[year >= 2010 & year <= 2015 & !
is.na(neet_share)]),
    baseline_year = last(year[year >= 2010 & year <= 2015 & !
is.na(neet_share)]),
    baseline_pop = last(population[year >= 2010 & year <= 2015 & !
is.na(population) & !is.na(neet_share)]),
    # Recent period (2016-2020)
    recent_neet = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      neet_share[year == 2020 & !is.na(neet_share)][1],
      last(neet_share[year >= 2016 & year < 2020 & !
is.na(neet_share)])
    ),
    recent_year = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      2020,
      last(year[year >= 2016 & year < 2020 & !is.na(neet_share)])
    ),
    recent_pop = ifelse(
      any(year == 2020 & !is.na(population) & !is.na(neet_share)),
      population[year == 2020 & !is.na(neet_share)][1],
      last(population[year >= 2016 & year < 2020 & !
is.na(neet_share)])
    ),
    .groups = 'drop'
  )
# REMOVED the filter here - keep all countries including those with NA

# Step 2: Calculate percentage change and determine target for each
country
country_target_achievement <- neet_population_weighted %>%
  mutate(
    pct_change_neet = ifelse(!is.na(baseline_neet) & !
is.na(recent_neet),
                             ((recent_neet - baseline_neet) /
baseline_neet) * 100,
                             NA_real_),
    # Determine target based on baseline NEET
    target_reduction = case_when(
      is.na(baseline_neet) ~ NA_real_,
      baseline_neet < 10 ~ -10,
      baseline_neet >= 10 & baseline_neet <= 30 ~ -20,
      baseline_neet > 30 ~ -10,
      TRUE ~ NA_real_
    ),
    # Determine if target was achieved
    target_status = case_when(
      is.na(baseline_neet) | is.na(recent_neet) | is.na(baseline_pop)
| is.na(recent_pop) ~ "No Data",
      pct_change_neet <= target_reduction ~ "Achieved",
      TRUE ~ "Missed"
    )
  )

# Step 3: For "No Data" countries, we need to estimate their
population
# Use the most recent available population data
country_target_achievement <- country_target_achievement %>%
  mutate(
    pop_for_calculation = coalesce(recent_pop, baseline_pop)
  )

# Step 3: Calculate total population by continent (including No Data
countries)
continent_total_pop <- country_target_achievement %>%
  group_by(continent) %>%
  reframe(total_continent_pop = sum(pop_for_calculation, na.rm =
TRUE), .groups = 'drop')

# Step 4: Calculate population-weighted percentages by target status
for each continent
population_weighted_summary <- country_target_achievement %>%
  left_join(continent_total_pop, by = "continent") %>%
  group_by(continent, target_status) %>%
  reframe(
    total_pop_in_category = sum(pop_for_calculation, na.rm = TRUE),
    total_continent_pop = first(total_continent_pop),
    .groups = 'drop'
  ) %>%
  mutate(
    percentage = (total_pop_in_category / total_continent_pop) * 100
  )

print("Population-weighted target achievement by continent:")
print(population_weighted_summary)

# Calculate population-weighted NEET for baseline and recent periods
by continent
continent_neet_weighted <- country_target_achievement %>%
  filter(!is.na(baseline_neet) & !is.na(recent_neet)) %>%  # Only
countries with both baseline and recent data
  group_by(continent) %>%
  reframe(
    # Population-weighted NEET values
    baseline_weighted_neet = sum(baseline_neet * baseline_pop, na.rm =
TRUE) / sum(baseline_pop, na.rm = TRUE),
    recent_weighted_neet = sum(recent_neet * recent_pop, na.rm = TRUE)
/ sum(recent_pop, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  mutate(
    # Calculate percentage change
    weighted_neet_change = ((recent_weighted_neet -
baseline_weighted_neet) / baseline_weighted_neet) * 100
  )

# Create bar chart
graph6 <- ggplot(continent_neet_weighted, aes(x = reorder(continent, -
weighted_neet_change),
    y = weighted_neet_change,
    fill = weighted_neet_change < 0)) +
  geom_col(alpha = 0.8) +
  geom_hline(yintercept = 0, linetype = "solid", color = "black") +
  geom_text(aes(label = paste0(round(weighted_neet_change, 1), "%")),
            vjust =
ifelse(continent_neet_weighted$weighted_neet_change < 0, 1.0, -0.5),
            fontface = "bold",
            size = 4) +
  scale_fill_manual(
    values = c("TRUE" = "#2166ac", "FALSE" = "#b2182b"),

  ) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey70"),
    panel.grid.minor = element_line(colour = "grey90", linewidth =
0.4),
    plot.title = element_text(hjust = 0.0, face = "bold", size = 16),
    plot.subtitle = element_text(size = 14),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none"
  ) +
  labs(
    x = "Continent",
    y = "Population-Weighted NEET Change (%)",
    fill = "NEET Trend",
    title = "Figure 7: Population-Weighted NEET Change",
    subtitle = "Baseline: 2010-2015 vs Recent: 2016-2020 (Latest
available)"
  )
```

```
# Print the table as well
print("Population-weighted NEET change by continent:")
print(continent_neet_weighted)

# Save to CSV
write_csv(continent_neet_weighted,
"output/continent_population_weighted_neet_change.csv")


# Calculate year-over-year NEET percentage change for each country
# Only for CONSECUTIVE years
yoy_neet_change <- master %>%
  filter(!is.na(neet_share) & !is.na(population)) %>%
  group_by(country, continent) %>%
  arrange(year) %>%
  mutate(
    # Check if this year and previous year are consecutive
    year_diff = year - lag(year),
    prev_neet = lag(neet_share),
    prev_pop = lag(population),
    # Calculate YoY percentage change ONLY if years are consecutive
    yoy_pct_change = ifelse(year_diff == 1 & !is.na(prev_neet),
                            ((neet_share - prev_neet) / prev_neet) *
100,
                            NA_real_)
  ) %>%
  filter(!is.na(yoy_pct_change)) %>%  # Keep only consecutive year
comparisons
  ungroup()

# Calculate global mean YoY percentage change (population-weighted)
# Only includes countries with consecutive year data
global_yoy_weighted <- yoy_neet_change %>%
  group_by(year) %>%
  reframe(
    total_pop = sum(population, na.rm = TRUE),
    weighted_yoy_pct_change = sum(yoy_pct_change * population, na.rm =
TRUE) / sum(population, na.rm = TRUE),
    n_countries = n(),
    n_unique_countries = n_distinct(country)
  )

# Overall global mean YoY percentage change across all years
global_mean_yoy_pct <- yoy_neet_change %>%
  reframe(
    # Simple average (unweighted)
    mean_yoy_pct_change_unweighted = mean(yoy_pct_change, na.rm =
TRUE),

    # Population-weighted average
    mean_yoy_pct_change_weighted = sum(yoy_pct_change * population,
na.rm = TRUE) / sum(population, na.rm = TRUE),

    median_yoy_pct_change = median(yoy_pct_change, na.rm = TRUE),
    sd_yoy_pct_change = sd(yoy_pct_change, na.rm = TRUE),
    n_consecutive_year_pairs = n(),
    n_countries_with_consecutive_data = n_distinct(country)
  )

print("Global mean year-over-year NEET percentage change (consecutive
years only):")
print(global_mean_yoy_pct)

print("\nGlobal YoY percentage change by year (population-weighted,
consecutive years only):")
print(global_yoy_weighted)

# Show which countries have consecutive year data
countries_with_consecutive <- yoy_neet_change %>%
  group_by(country, continent) %>%
  reframe(
    n_consecutive_pairs = n(),
    mean_yoy_pct = mean(yoy_pct_change, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  arrange(desc(n_consecutive_pairs))

print("\nCountries with consecutive year data:")
print(head(countries_with_consecutive, 20))

# Calculate by continent (consecutive years only)
continent_yoy_pct <- yoy_neet_change %>%
  group_by(continent) %>%
  reframe(
    mean_yoy_pct_change_weighted = sum(yoy_pct_change * population,
na.rm = TRUE)
      / sum(population, na.rm = TRUE),
    mean_yoy_pct_change_unweighted = mean(yoy_pct_change, na.rm =
TRUE),
    n_consecutive_pairs = n(),
    n_countries = n_distinct(country),
    .groups = 'drop'
  ) %>%
  arrange(mean_yoy_pct_change_weighted)

print("\nMean YoY NEET percentage change by continent (consecutive
years only):")
print(continent_yoy_pct)
```

```
# Save results
write_csv(global_mean_yoy_pct,
"output/global_mean_yoy_neet_pct_change_consecutive.csv")
write_csv(global_yoy_weighted,
"output/global_yoy_pct_by_year_consecutive.csv")
write_csv(continent_yoy_pct,
"output/continent_mean_yoy_neet_pct_change_consecutive.csv")
write_csv(countries_with_consecutive,
"output/countries_with_consecutive_year_data.csv")

# Calculate percentage change in NEET share for each country
# Using earliest available year between 2010-2015 as baseline
# and 2020 or closest year before 2020 (2016-2019) as endpoint
neet_country_change <- master %>%
  filter(!is.na(neet_share)) %>%
  group_by(country) %>%
  arrange(year) %>%
  reframe(
    baseline_neet = first(neet_share[year >= 2010 & year <= 2015]),
    baseline_year = first(year[year >= 2010 & year <= 2015]),
    recent_neet = ifelse(
      any(year == 2018 & !is.na(neet_share)),
      neet_share[year == 2018][1],
      last(neet_share[year >= 2016 & year < 2018])
    ),
    recent_year = ifelse(
      any(year == 2018 & !is.na(neet_share)),
      2018,
      last(year[year >= 2016 & year < 2018])
    )
  ) %>%
  filter(!is.na(baseline_neet) & !is.na(recent_neet)) %>%
  mutate(neet_pct_change = ((recent_neet - baseline_neet) /
baseline_neet) * 100)

# Get world map data
world_map <- map_data("world")

# Match country names between datasets
neet_country_change <- neet_country_change %>%
  mutate(region = case_when(
    country == "United States" ~ "USA",
    country == "Czechia" ~ "Czech Republic",
    country == "Hong Kong" ~ "China",
    TRUE ~ country
  ))

# Join the data
map_data_joined <- world_map %>%
  left_join(neet_country_change, by = "region")
```

```
# Create the map
graph7 <- ggplot(map_data_joined, aes(x = long, y = lat, group =
group, fill = neet_pct_change)) +
  geom_polygon(color = "white", linewidth = 0.1) +
  scale_fill_gradient2(
    low = "#2166ac",
    mid = "white",
    high = "#b2182b",
    midpoint = 0,
    limits = c(-50, 50),
    na.value = "grey80",
    name = "NEET Change (%)",
    oob = scales::squish
  ) +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    axis.text = element_blank(),
    axis.title = element_blank(),
    axis.ticks = element_blank(),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.key.width = unit(2, "cm"),
    plot.title = element_text(hjust = 0.0, face = "bold")
  ) +
  coord_fixed(1.3) +
  labs(title = "Figure 8: Percentage Change in Youth NEET Share",
       subtitle = "Country Level Comparison (EXCLUDING COVID YEARS)")



# Step 1: Get ALL countries from master dataset (2010-2021) with their
most recent population
all_countries_with_pop <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  group_by(country, continent) %>%
  arrange(desc(year)) %>%
  reframe(
    most_recent_pop = first(population[!is.na(population)]),
    .groups = 'drop'
  ) %>%
  filter(!is.na(most_recent_pop))  # Only keep countries with at least
some population data

# Step 2: Calculate NEET change for countries WITH data
neet_country_change <- master %>%
  filter(!is.na(neet_share)) %>%
  group_by(country, continent) %>%
  arrange(year) %>%
  reframe(
    baseline_neet = first(neet_share[year >= 2010 & year <= 2015]),
    baseline_year = first(year[year >= 2010 & year <= 2015]),
    baseline_pop = first(population[year >= 2010 & year <= 2015 & !
is.na(population)]),
    recent_neet = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      neet_share[year == 2020][1],
      last(neet_share[year >= 2016 & year < 2020])
    ),
    recent_year = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      2020,
      last(year[year >= 2016 & year < 2020])
    ),
    recent_pop = ifelse(
      any(year == 2020 & !is.na(population)),
      population[year == 2020][1],
      last(population[year >= 2016 & year < 2020 & !
is.na(population)])
    ),
    .groups = 'drop'
  ) %>%
  mutate(
    pct_change_neet = ifelse(!is.na(baseline_neet) & !
is.na(recent_neet),
                             ((recent_neet - baseline_neet) /
baseline_neet) * 100,
                             NA_real_),
    target_reduction = case_when(
      is.na(baseline_neet) ~ NA_real_,
      baseline_neet < 10 ~ -10,
      baseline_neet >= 10 & baseline_neet <= 30 ~ -20,
      baseline_neet > 30 ~ -10,
      TRUE ~ NA_real_
    ),
    target_status = case_when(
      is.na(baseline_neet) | is.na(recent_neet) ~ "No Data",
      pct_change_neet <= target_reduction ~ "Achieved",
      TRUE ~ "Missed"
    )
  )

# Step 3: Join ALL countries with NEET change results
countries_complete <- all_countries_with_pop %>%
  left_join(
    neet_country_change %>% select(country, target_status,
recent_pop),
    by = "country"
  ) %>%
  mutate(
    target_status = ifelse(is.na(target_status), "No Data",
target_status),
    pop_for_calculation = coalesce(recent_pop, most_recent_pop)
  )

# Step 4: Calculate total population by continent (including No Data
countries)
continent_total_pop <- countries_complete %>%
  group_by(continent) %>%
  mutate(total_continent_pop = sum(pop_for_calculation, na.rm = TRUE))

# Step 5: Calculate population-weighted percentages INCLUDING "No
Data"
population_weighted_summary_complete <- countries_complete %>%
  left_join(continent_total_pop, by = "country") %>%
  group_by(continent.x, target_status.x) %>%
  reframe(
    total_pop_in_category = sum(pop_for_calculation.x, na.rm = TRUE),
    total_continent_pop = first(total_continent_pop),
    .groups = 'drop'
  ) %>%
  mutate(
    percentage = (total_pop_in_category / total_continent_pop) * 100
  )

print(population_weighted_summary_complete)

# Step 6: Create stacked bar chart with all three categories

# Calculate percentage of NA for NEET data (2010-2021)
neet_na_analysis <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  group_by(continent) %>%
  reframe(
    total_observations = n(),
    neet_na_count = sum(is.na(neet_share)),
    neet_available_count = sum(!is.na(neet_share)),
    neet_na_percentage = round((sum(is.na(neet_share)) / n()) * 100,
1),
    neet_available_percentage = round((sum(!is.na(neet_share)) / n())
* 100, 1),
    .groups = 'drop'
  )


# Overall global statistics
global_neet_na <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  reframe(
    total_observations = n(),
    neet_na_count = sum(is.na(neet_share)),
    neet_available_count = sum(!is.na(neet_share)),
    neet_na_percentage = round((sum(is.na(neet_share)) / n()) * 100,
1),
    neet_available_percentage = round((sum(!is.na(neet_share)) / n())
* 100, 1)
  )

print("\nGlobal NEET data availability (2010-2021):")
print(global_neet_na)

# By year
neet_na_by_year <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  group_by(year) %>%
  reframe(
    total_countries = n(),
    neet_na_count = sum(is.na(neet_share)),
    neet_available_count = sum(!is.na(neet_share)),
    neet_na_percentage = round((sum(is.na(neet_share)) / n()) * 100,
1),
    .groups = 'drop'
  )

print("\nNEET data availability by year (2010-2021):")
print(neet_na_by_year)

# Calculate how close each country is to their target
# Filter for countries that DON'T have 2015 OR DON'T have 2020
closest_5_countries_missing_data <- country_target_achievement %>%
  filter(!is.na(pct_change_neet) & !is.na(target_reduction)) %>%
  filter(baseline_year != 2015 | recent_year != 2020) %>%  # Missing
2015 OR 2020
  mutate(
    # Calculate distance from target (negative = exceeded target,
positive = fell short)
    distance_from_target = pct_change_neet - target_reduction,
    abs_distance_from_target = abs(distance_from_target)
  ) %>%
  arrange(abs_distance_from_target) %>%
  select(country, continent, baseline_year, recent_year,
baseline_neet, recent_neet,
         pct_change_neet, target_reduction, distance_from_target,
target_status) %>%
  head(5)

print("Top 5 countries closest to their target (missing 2015 OR 2020
data):")
```

```
# Find countries closest to 10% or 30% NEET
countries_near_thresholds <- country_target_achievement %>%
  filter(!is.na(baseline_neet)) %>%
  mutate(
    # Calculate distance from 10% and 30%
    distance_from_10 = abs(baseline_neet - 10),
    distance_from_30 = abs(baseline_neet - 30),
    # Which threshold is closer?
    closest_threshold = ifelse(distance_from_10 < distance_from_30,
"10%", "30%"),
    distance_from_threshold = pmin(distance_from_10, distance_from_30)
  ) %>%
  arrange(distance_from_threshold) %>%
  select(country, continent, baseline_year, baseline_neet,
closest_threshold, distance_from_threshold) %>%
  head(10)

# Separate by threshold
closest_to_10 <- country_target_achievement %>%
  filter(!is.na(baseline_neet)) %>%
  mutate(distance_from_10 = abs(baseline_neet - 10)) %>%
  arrange(distance_from_10) %>%
  select(country, continent, baseline_year, baseline_neet,
distance_from_10) %>%
  head(5)

print("\nTop 5 countries closest to 10% NEET:")
print(closest_to_10)

closest_to_30 <- country_target_achievement %>%
  filter(!is.na(baseline_neet)) %>%
  mutate(distance_from_30 = abs(baseline_neet - 30)) %>%
  arrange(distance_from_30) %>%
  select(country, continent, baseline_year, baseline_neet,
distance_from_30) %>%
  head(5)

print("\nTop 5 countries closest to 30% NEET:")
print(closest_to_30)

# Save
write_csv(countries_near_thresholds,
"output/countries_near_10_or_30_threshold.csv")
write_csv(closest_to_10, "output/countries_closest_to_10_percent.csv")
write_csv(closest_to_30, "output/countries_closest_to_30_percent.csv")

# Calculate percentage of NA for NEET data (2010-2021)
neet_na_analysis <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  group_by(continent) %>%
  reframe(
    total_observations = n(),
    neet_na_count = sum(is.na(neet_share)),
    neet_available_count = sum(!is.na(neet_share)),
    neet_na_percentage = round((sum(is.na(neet_share)) / n()) * 100,
1),
    neet_available_percentage = round((sum(!is.na(neet_share)) / n())
* 100, 1),
    .groups = 'drop'
  )

# Overall global statistics
global_neet_na <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  reframe(
    total_observations = n(),
    neet_na_count = sum(is.na(neet_share)),
    neet_available_count = sum(!is.na(neet_share)),
    neet_na_percentage = round((sum(is.na(neet_share)) / n()) * 100,
1),
    neet_available_percentage = round((sum(!is.na(neet_share)) / n())
* 100, 1)
  )

print("\nGlobal NEET data availability (2010-2021):")
print(global_neet_na)

# By year
neet_na_by_year <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  group_by(year) %>%
  reframe(
    total_countries = n(),
    neet_na_count = sum(is.na(neet_share)),
    neet_available_count = sum(!is.na(neet_share)),
    neet_na_percentage = round((sum(is.na(neet_share)) / n()) * 100,
1),
    .groups = 'drop'
  )

# Prepare data for box plot - remove missing NEET values
neet_continent_data <- master_data %>%
  filter(!is.na(neet_share), !is.na(continent))

# Create the box plot
graph8 <- ggplot(neet_continent_data, aes(x = continent, y =
neet_share, fill = continent)) +
  geom_boxplot(outlier.shape = 16, outlier.size = 2, alpha = 0.7) +
  # Customize colors to match the reference style
  scale_fill_manual(values = c(
    "Africa" = "#D55E00",
    "Asia" = "#E6AB02",
    "Europe" = "#009E73",
    "North America" = "#56B4E9",
    "Oceania" = "#0072B2",
    "South America" = "#CC79A7"
  )) +

  # Labels and theme
  labs(
    title = "Box Plot: NEET Distribution by Continent",
    x = "Continent",
    y = "Share of Youth Not in Education,\nEmployment or Training
(%)",
    fill = "continent"
  ) +

  # Clean theme matching the reference style
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 10),
    legend.position = "bottom",
    legend.title = element_text(size = 10, face = "bold"),
    legend.text = element_text(size = 9),
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, linewidth
= 0.5)
  ) +

  # Arrange legend items horizontally
  guides(fill = guide_legend(nrow = 1))

# Save the plot
# ggsave("output/neet_boxplot_by_continent.png", width = 10, height =
7, dpi = 300)

# Print summary statistics
cat("\n=== NEET Distribution Summary by Continent ===\n\n")
neet_summary <- neet_continent_data %>%
  group_by(continent) %>%
  reframe(
    n_observations = n(),
    n_countries = n_distinct(code),
    min_neet = round(min(neet_share, na.rm = TRUE), 2),
    q1_neet = round(quantile(neet_share, 0.25, na.rm = TRUE), 2),
    median_neet = round(median(neet_share, na.rm = TRUE), 2),
    q3_neet = round(quantile(neet_share, 0.75, na.rm = TRUE), 2),
    max_neet = round(max(neet_share, na.rm = TRUE), 2),
    mean_neet = round(mean(neet_share, na.rm = TRUE), 2),
    .groups = "drop"
  )

print(neet_summary)




#
======================================================================
======
# NEET Trends Over Time by Continent
#
======================================================================
======

# Prepare data: calculate mean NEET by continent and year
neet_time_data <- master_data %>%
  filter(!is.na(neet_share), !is.na(continent)) %>%
  group_by(continent, year) %>%
  reframe(
    mean_neet = mean(neet_share, na.rm = TRUE),
    median_neet = median(neet_share, na.rm = TRUE),
    n_countries = n_distinct(code),
    .groups = "drop"
  )

cat("\n=== NEET Trends Over Time Created ===\n")
cat("✓ Saved: neet_trends_over_time.png\n\n")

#
======================================================================
======
# BONUS: Create a faceted plot for detailed view by continent
#
======================================================================
======

# Prepare individual country data
neet_country_time <- master_data %>%
  filter(!is.na(neet_share), !is.na(continent))

#
```

```
======================================================================
======
# Print summary statistics
#
======================================================================
======

cat("=== NEET Change Over Time Summary ===\n\n")

# Calculate first and last year averages by continent
time_summary <- neet_time_data %>%
  group_by(continent) %>%
  arrange(year) %>%
  filter(year == min(year) | year == max(year)) %>%
  reframe(
    year_range = paste(min(year), "-", max(year)),
    initial_neet = first(mean_neet),
    final_neet = last(mean_neet),
    absolute_change = final_neet - initial_neet,
    percent_change = round((final_neet - initial_neet) / initial_neet
* 100, 1),
    .groups = "drop"
  )

print(time_summary)

cat("\n✓ All visualizations created successfully!\n")

cat("\n✓ Box plot saved as: neet_boxplot_by_continent.png\n")



#
======================================================================
======
# NEET vs Education Spending - Interactive Visualization
#
======================================================================
======

# Prepare data: filter for complete cases
neet_edu_data <- master_data %>%
  filter(!is.na(neet_share), !is.na(edu_expenditure_gdp), !
is.na(continent)) %>%
  # Get most recent year for each country to avoid overplotting
  group_by(code) %>%
  arrange(desc(year)) %>%
  slice(1) %>%
  ungroup()
```

```
# Identify interesting countries to label (extremes and notable cases)
countries_to_label <- neet_edu_data %>%
  group_by(continent) %>%
  arrange(desc(neet_share)) %>%
  slice(1:2) %>%  # Top 2 highest NEET per continent
  ungroup() %>%
  bind_rows(
    neet_edu_data %>%
      filter(edu_expenditure_gdp > 6 | neet_share > 25)  # High
spenders or high NEET
  ) %>%
  distinct(code, .keep_all = TRUE)

#
======================================================================
======
# CALCULATE CORRELATION STATISTICS
#
======================================================================
======

cat("=== CORRELATION ANALYSIS ===\n\n")

# Overall correlation
overall_cor <- cor(neet_edu_data$edu_expenditure_gdp,
                   neet_edu_data$neet_share,
                   use = "complete.obs")
cat("Overall Correlation:", round(overall_cor, 3), "\n\n")

# Correlation by continent
continent_cor <- neet_edu_data %>%
  group_by(continent) %>%
  reframe(
    n_countries = n(),
    correlation = round(cor(edu_expenditure_gdp, neet_share,
                            use = "complete.obs"), 3),
    mean_edu_spending = round(mean(edu_expenditure_gdp, na.rm = TRUE),
2),
    mean_neet = round(mean(neet_share, na.rm = TRUE), 2),
    .groups = "drop"
  ) %>%
  arrange(desc(abs(correlation)))

print(continent_cor)

# Linear regression summary
cat("\n=== LINEAR REGRESSION MODEL ===\n")
model <- lm(neet_share ~ edu_expenditure_gdp, data = neet_edu_data)
```

```
print(summary(model))

cat("\n✓ All visualizations created successfully!\n")
cat("\nKEY INSIGHTS:\n")
cat("- Point size = GDP per capita (bigger = wealthier)\n")
cat("- Point color = Continent\n")
cat("- Gray line = Overall trend\n")
cat("- Labels = Notable high-NEET or high-spending countries\n")


#
======================================================================
======
# NEET vs GDP Growth Rate - Clean & Simple Visualization
#
======================================================================
======

#
======================================================================
======
# Calculate GDP Growth Rate
#
======================================================================
======

cat("=== Calculating GDP Growth Rates ===\n")

# Calculate year-over-year GDP growth rate
gdp_growth_data <- master_data %>%
  filter(!is.na(gdp_per_capita)) %>%
  arrange(code, year) %>%
  group_by(code) %>%
  mutate(
    gdp_growth_rate = (gdp_per_capita - lag(gdp_per_capita)) /
lag(gdp_per_capita) * 100
  ) %>%
  ungroup()

# Combine with NEET data
neet_gdp_growth <- gdp_growth_data %>%
  filter(!is.na(neet_share), !is.na(gdp_growth_rate), !
is.na(continent)) %>%
  filter(gdp_growth_rate > -30, gdp_growth_rate < 30)  # Remove
extreme outliers

cat("Total observations:", nrow(neet_gdp_growth), "\n\n")

#
```

```
=======================================================================
======
# Simple Statistics
#
=======================================================================
======

cat("=== KEY FINDINGS ===\n\n")

# Overall correlation
overall_cor <- cor(neet_gdp_growth$gdp_growth_rate,
                   neet_gdp_growth$neet_share,
                   use = "complete.obs")
cat("Correlation between GDP growth and NEET:", round(overall_cor, 3),
"\n")

if(overall_cor < -0.1) {
  cat("→ Negative relationship: Higher growth = Lower NEET\n\n")
} else if(overall_cor > 0.1) {
  cat("→ Positive relationship: Higher growth = Higher NEET\n\n")
} else {
  cat("→ Weak/No clear relationship\n\n")
}

# Simple summary by economic condition
cat("Average NEET by Economic Condition:\n")
growth_summary <- neet_gdp_growth %>%
  mutate(
    condition = case_when(
      gdp_growth_rate < 0 ~ "Negative Growth",
      gdp_growth_rate < 3 ~ "Low Growth (0-3%)",
      TRUE ~ "Strong Growth (>3%)"
    )
  ) %>%
  group_by(condition) %>%
  reframe(
    observations = n(),
    avg_neet = round(mean(neet_share), 1),
    .groups = "drop"
  )

print(growth_summary)

cat("\n✓ Clean visualization created!\n")

print("\nNEET data availability by year (2010-2021):")
print(neet_na_by_year)

# List of LDCs (as of recent UN classification)
```

```
ldc_countries <- c(
  "Afghanistan", "Angola", "Bangladesh", "Benin", "Bhutan", "Burkina
Faso",
  "Burundi", "Cambodia", "Central African Republic", "Chad",
"Comoros",
  "Congo, Dem. Rep.", "Djibouti", "Eritrea", "Ethiopia", "Gambia",
"Guinea",
  "Guinea-Bissau", "Haiti", "Kiribati", "Laos", "Lesotho", "Liberia",
  "Madagascar", "Malawi", "Mali", "Mauritania", "Mozambique",
"Myanmar",
  "Nepal", "Niger", "Rwanda", "Sao Tome and Principe", "Senegal",
  "Sierra Leone", "Solomon Islands", "Somalia", "South Sudan",
"Sudan",
  "Tanzania", "Timor-Leste", "Togo", "Tuvalu", "Uganda", "Yemen",
"Zambia"
)

# Calculate GDP growth rates and identify LDC status
gdp_growth_data <- master %>%
  filter(!is.na(gdp_per_capita)) %>%
  mutate(is_ldc = country %in% ldc_countries) %>%
  group_by(country, continent, is_ldc) %>%
  arrange(year) %>%
  mutate(
    # Calculate year-over-year GDP growth rate
    year_diff = year - lag(year),
    gdp_growth = ifelse(year_diff == 1,
                        ((gdp_per_capita - lag(gdp_per_capita)) /
lag(gdp_per_capita)) * 100,
                        NA_real_)
  ) %>%
  filter(!is.na(gdp_growth)) %>%
  ungroup()

# Calculate baseline and recent NEET for each country
neet_boxplot_data <- master %>%
  group_by(country, continent) %>%
  arrange(year) %>%
  reframe(
    # Baseline period (2010-2015) - take last available year
    baseline_neet = last(neet_share[year >= 2010 & year <= 2015 & !
is.na(neet_share)]),
    baseline_year = last(year[year >= 2010 & year <= 2015 & !
is.na(neet_share)]),
    # Recent period (2016-2020) - prioritize 2020, then take last
available
    recent_neet = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      neet_share[year == 2020 & !is.na(neet_share)][1],
      last(neet_share[year >= 2016 & year < 2020 & !
is.na(neet_share)])
    ),
    recent_year = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      2020,
      last(year[year >= 2016 & year < 2020 & !is.na(neet_share)])
    ),
    .groups = 'drop'
  ) %>%
  filter(!is.na(baseline_neet) & !is.na(recent_neet))


# Plot 1: Baseline NEET variance by continent
graph9 <- ggplot(neet_boxplot_data, aes(x = continent, y =
baseline_neet, fill = continent)) +
  geom_boxplot(alpha = 0.7) +
  coord_cartesian(ylim = c(0, 70)) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey70"),
    panel.grid.minor = element_line(colour = "grey90", linewidth =
0.4),
    legend.position = "none",
    plot.title = element_text(hjust = 0, face = "bold"),
    plot.subtitle = element_text(hjust = 0),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    x = "Continent",
    y = "Youth NEET Rate (%)",
    title = "Figure 10: Variance of Youth NEET Rates by Continent",
    subtitle = "Baseline Period: 2010-2015 (Last Available Year)"
  )

# Plot 2: Recent NEET variance by continent
graph10 <- ggplot(neet_boxplot_data, aes(x = continent, y =
recent_neet, fill = continent)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey70"),
    panel.grid.minor = element_line(colour = "grey90", linewidth =
0.4),
    legend.position = "none",
    plot.title = element_text(hjust = 0, face = "bold"),
    plot.subtitle = element_text(hjust = 0),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    x = "Continent",
    y = "Youth NEET Rate (%)",
    title = "Figure 11: Variance of Youth NEET Rates by Continent",
    subtitle = "Comparison Period: 2016-2020 (Prioritize 2020, Then 
Last Available)"
  )

# Optional: Combined plot showing both periods side by side
library(tidyr)

neet_combined <- neet_boxplot_data %>%
  select(country, continent, baseline_neet, recent_neet) %>%
  pivot_longer(
    cols = c(baseline_neet, recent_neet),
    names_to = "period",
    values_to = "neet_rate"
  ) %>%
  mutate(
    period = ifelse(period == "baseline_neet", "Baseline (2010-2015)", 
"Recent (2016-2020)")
  )

# Calculate NEET change for each country
neet_change_data <- master %>%
  group_by(country, continent) %>%
  arrange(year) %>%
  reframe(
    # Baseline period (2010-2015) - take last available year
    baseline_neet = last(neet_share[year >= 2010 & year <= 2015 & !
is.na(neet_share)]),
    # Recent period (2016-2020) - prioritize 2020, then take last 
available
    recent_neet = ifelse(
      any(year == 2020 & !is.na(neet_share)),
      neet_share[year == 2020 & !is.na(neet_share)][1],
      last(neet_share[year >= 2016 & year < 2020 & !
is.na(neet_share)])
    ),
    .groups = 'drop'
  ) %>%
  filter(!is.na(baseline_neet) & !is.na(recent_neet)) %>%
  mutate(
    neet_change = ((recent_neet - baseline_neet) / baseline_neet) * 
100
  )

# Calculate median for each continent
median_values <- neet_change_data %>%
  group_by(continent) %>%
  reframe(median_change = median(neet_change, na.rm = TRUE), .groups =
'drop')

# Print median values
print("Median NEET change by continent:")
print(median_values)

#NEW PLOT: DOT PLOT
neet_ranges <- neet_change_data %>%
  mutate(
    baseline_range = cut(
      baseline_neet,
      breaks = c(-Inf, 10, 20, 30, 40, Inf),
      labels = c("<10", "10–20", "20–30", "30–40", ">40"),
      right = FALSE
    ),
    recent_range = cut(
      recent_neet,
      breaks = c(-Inf, 10, 20, 30, 40, Inf),
      labels = c("<10", "10–20", "20–30", "30–40", ">40"),
      right = FALSE
    )
  )

baseline_counts <- neet_ranges %>%
  filter(baseline_range %in% c("<10", ">40")) %>%
  group_by(continent, baseline_range) %>%
  reframe(n = n(), .groups = "drop")

recent_counts <- neet_ranges %>%
  filter(recent_range %in% c("<10", ">40")) %>%
  group_by(continent, recent_range) %>%
  reframe(n = n(), .groups = "drop")

common_breaks <- c(1,5,10,20)

baseline_counts_filtered <- baseline_counts %>%
  filter(continent != "North America")



graph11 <- ggplot(baseline_counts_filtered, aes(y = baseline_range, x
= continent)) +
  geom_point(aes(size = n, color = continent)) +
  scale_size_continuous(range = c(3, 12),
                        breaks = common_breaks,
                        limits = c(0,max(common_breaks))
  ) +
  guides(color = "none") +   # ← removes continent colour legend
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.2),
    plot.subtitle = element_text(face = "bold")
  ) +
  labs(
    title = "Figure 12a: Distribution of Baseline NEET Levels by
Continent (2010–2015)",
    x = "Continent",
    y = "Baseline NEET Range (%)",
    size = "Number of Countries"
  )

graph12 <- ggplot(recent_counts, aes(x = continent, y = recent_range))
+
  geom_point(aes(size = n, color = continent)) +
  scale_size_continuous(range = c(3, 12),
                        breaks = common_breaks,
                        limits = c(0,max(common_breaks))
  ) +
  guides(color = "none") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.2),
    plot.subtitle = element_text(face = "bold")
  ) +
  labs(
    title = "Figure 12b: Distribution of Comparison NEET Levels by
Continent (2016–2020)",
    x = "Continent",
    y = "Recent NEET Range (%)",
    size = "Number of Countries"
  )

population_weighted_summary_complete <-
population_weighted_summary_complete %>%
  rename(target_status = target_status.x)
graph5 <- ggplot(population_weighted_summary_complete, aes(x =
continent.x, y = percentage, fill = target_status)) +
  geom_col(position = "stack", alpha = 0.8) +
  scale_fill_manual(
    values = c(
      "Achieved" = "#2166ac",
      "Missed" = "#b2182b",
      "No Data" = "grey80"
    )
  ) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey70"),
    panel.grid.minor = element_line(colour = "grey90", linewidth =
0.4),
    legend.position = "bottom",
    legend.direction = "horizontal",
    plot.title = element_text(hjust = 0, face = "bold", size = 16),
    plot.subtitle = element_text(size = 14),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    x = "Continent",
    y = "Percentage of Total Population (%)",
    fill = "Target Achievement",
    title = "Figure 9: NEET Reduction Target Achievement",
    subtitle = "Population Weighted Continent level comparison"
  ) +
  geom_text(
    aes(label = ifelse(percentage > 5, paste0(round(percentage, 1),
"%"), "")),
    position = position_stack(vjust = 0.5),
    color = "white",
    fontface = "bold",
    size = 3
  )

neet_country_change <- master %>%
  filter(!is.na(neet_share)) %>%
  group_by(country) %>%
  arrange(year) %>%
  reframe(
    baseline_neet = first(neet_share[year >= 2010 & year <= 2015]),
    baseline_year = first(year[year >= 2010 & year <= 2015]),
    recent_neet = ifelse(
      any(year == 2018 & !is.na(neet_share)),
      neet_share[year == 2018][1],
      last(neet_share[year >= 2016 & year < 2018])
    ),
    recent_year = ifelse(
      any(year == 2018 & !is.na(neet_share)),
      2018,
      last(year[year >= 2016 & year < 2018])
    )
  ) %>%
  filter(!is.na(baseline_neet) & !is.na(recent_neet)) %>%
  mutate(neet_pct_change = ((recent_neet - baseline_neet) /
baseline_neet) * 100)

# Get world map data
world_map <- map_data("world")

# Match country names between datasets
```

```
neet_country_change <- neet_country_change %>%
  mutate(region = case_when(
    country == "United States" ~ "USA",
    country == "Czechia" ~ "Czech Republic",
    country == "Hong Kong" ~ "China",
    TRUE ~ country
  ))

# Join the data
map_data_joined <- world_map %>%
  left_join(neet_country_change, by = "region")

# Create the map
graph13 <- ggplot(map_data_joined, aes(x = long, y = lat, group =
group, fill = neet_pct_change)) +
  geom_polygon(color = "white", linewidth = 0.1) +
  scale_fill_gradient2(
    low = "#2166ac",
    mid = "white",
    high = "#b2182b",
    midpoint = 0,
    limits = c(-50, 50),
    na.value = "grey80",
    name = "NEET Change (%)",
    oob = scales::squish
  ) +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    axis.text = element_blank(),
    axis.title = element_blank(),
    axis.ticks = element_blank(),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.key.width = unit(2, "cm"),
    plot.title = element_text(hjust = 0.0, face = "bold")
  ) +
  coord_fixed(1.3) +
  labs(title = "Figure 8: Percentage Change in Youth NEET Share",
       subtitle = "Country Level Comparison (EXCLUDING COVID YEARS)")

# Step 1: Get ALL countries from master dataset (2010-2021) with their
most recent population
all_countries_with_pop <- master %>%
  filter(year >= 2010 & year <= 2021) %>%
  group_by(country, continent) %>%
  arrange(desc(year)) %>%
  reframe(
    most_recent_pop = first(population[!is.na(population)])
  ) %>%
  filter(!is.na(most_recent_pop))  # Only keep countries with at least 
some population data

# Step 2: Calculate NEET change for countries WITH data
neet_country_change <- master %>%
  filter(!is.na(neet_share)) %>%
  group_by(country, continent) %>%
  arrange(year) %>%
  reframe(
    baseline_neet = first(neet_share[year >= 2010 & year <= 2015]),
    baseline_year = first(year[year >= 2010 & year <= 2015]),
    baseline_pop = first(population[year >= 2010 & year <= 2015 & !
is.na(population)]),
    recent_neet = ifelse(
      any(year == 2018 & !is.na(neet_share)),
      neet_share[year == 2018][1],
      last(neet_share[year >= 2016 & year < 2018])
    ),
    recent_year = ifelse(
      any(year == 2018 & !is.na(neet_share)),
      2018,
      last(year[year >= 2016 & year < 2018])
    ),
    recent_pop = ifelse(
      any(year == 2018 & !is.na(population)),
      population[year == 2018][1],
      last(population[year >= 2016 & year < 2018 & !
is.na(population)])
    )
  ) %>%
  mutate(
    pct_change_neet = ifelse(!is.na(baseline_neet) & !
is.na(recent_neet),
                             ((recent_neet - baseline_neet) / 
baseline_neet) * 100,
                             NA_real_),
    target_reduction = case_when(
      is.na(baseline_neet) ~ NA_real_,
      baseline_neet < 10 ~ -10,
      baseline_neet >= 10 & baseline_neet <= 30 ~ -20,
      baseline_neet > 30 ~ -10,
      TRUE ~ NA_real_
    ),
    target_status = case_when(
      is.na(baseline_neet) | is.na(recent_neet) ~ "No Data",
      pct_change_neet <= target_reduction ~ "Achieved",
      TRUE ~ "Missed"
    )
  )
```

```r
# Step 3: Join ALL countries with NEET change results
countries_complete <- all_countries_with_pop %>%
  left_join(
    neet_country_change %>% select(country, target_status,
recent_pop),
    by = "country"
  ) %>%
  mutate(
    target_status = ifelse(is.na(target_status), "No Data",
target_status),
    pop_for_calculation = coalesce(recent_pop, most_recent_pop)
  )

# Step 4: Calculate total population by continent (including No Data
countries)
continent_total_pop <- countries_complete %>%
  group_by(continent) %>%
  reframe(total_continent_pop = sum(pop_for_calculation, na.rm =
TRUE))

# Step 5: Calculate population-weighted percentages INCLUDING "No
Data"
population_weighted_summary_complete <- countries_complete %>%
  left_join(continent_total_pop, by = "continent") %>%
  group_by(continent, target_status) %>%
  reframe(
    total_pop_in_category = sum(pop_for_calculation, na.rm = TRUE),
    total_continent_pop = first(total_continent_pop)
  ) %>%
  mutate(
    percentage = (total_pop_in_category / total_continent_pop) * 100
  )

print(population_weighted_summary_complete)

# Step 6: Create stacked bar chart with all three categories
graph14 <- ggplot(population_weighted_summary_complete, aes(x =
continent, y = percentage, fill = target_status)) +
  geom_col(position = "stack", alpha = 0.8) +
  scale_fill_manual(
    values = c(
      "Achieved" = "#2166ac",
      "Missed" = "#b2182b",
      "No Data" = "grey80"
    )
  ) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey70"),
    panel.grid.minor = element_line(colour = "grey90", linewidth =
0.4),
    legend.position = "bottom",
    legend.direction = "horizontal",
    plot.title = element_text(hjust = 0, face = "bold", size = 16),
    plot.subtitle = element_text(size = 14),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    x = "Continent",
    y = "Percentage of Total Population (%)",
    fill = "Target Achievement",
    title = "Figure 9: NEET Reduction Target Achievement",
    subtitle = "Population Weighted Continent level comparison"
  ) +
  geom_text(
    aes(label = ifelse(percentage > 5, paste0(round(percentage, 1),
"%"), "")),
    position = position_stack(vjust = 0.5),
    color = "white",
    fontface = "bold",
    size = 3
  )

figure_3 <- graph1
figure_4 <- graph2
figure_5 <- graph3
figure_6 <- graph4
figure_7 <- graph6
figure_8 <- graph7
figure_8b <- graph13
figure_9 <- graph5
figure_9b <- graph14
figure_10 <- graph9
figure_11 <- graph10
figure_12a <- graph11
figure_12b <- graph12
```