

Data Science: Group 23

2024-12-02

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v forcats   1.0.0      v stringr   1.5.1
## v lubridate 1.9.3      v tibble   3.2.1
## v purrr     1.0.2      v tidyr    1.3.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(ggmap)
library(rnaturalearth)
```

```
# Load datasets
continents <- read_csv("continents-according-to-our-world-in-data.csv")
```

```
## Rows: 285 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr (3): Entity, Code, Continent
```

```
## dbl (1): Year
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
gdp <- read_csv("gdp-per-capita-worldbank.csv")
```

```
## Rows: 6346 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, GDP per capita, PPP (constant 2017 international $)
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
neet_data <- read_csv("youth-not-in-education-employment-training.csv")
neet_data <- neet_data %>%
  rename(
    Share = `Share.of.youth.not.in.education..employment.or.training..total....of.youth.population.`
  ) #Rename column
```

```
##Target 1
```

```
# Merge GDP and continent datasets by the common column 'Entity' (country names)
merged_data <- gdp %>%
  inner_join(
    continents %>% select(Entity, Continent), # Select only relevant columns from 'continents'
    by = "Entity" # Match rows by the 'Entity' column
  )

# Filter to exclude Antarctica as it's not relevant for the analysis
filtered_data <- merged_data %>%
  filter(Continent != "Antarctica")

# Rename the GDP column for easier use
filtered_data <- filtered_data %>%
  rename(
    GDP_per_capita_PPP = `GDP per capita, PPP (constant 2017 international $)` # Simplify column name
  )
```

```
# Calculate GDP growth for each country by year
filtered_data <- filtered_data %>%
  arrange(Entity, Year) %>% # Ensure data is sorted by country and year
  group_by(Entity) %>% # Group by country for yearly calculations
  mutate(
    GDP_Growth = (GDP_per_capita_PPP - lag(GDP_per_capita_PPP)) / lag(GDP_per_capita_PPP) * 100
    # GDP growth is calculated as the percentage change from the previous year
  )
```

```
# Select relevant columns for the final dataset
final_data <- filtered_data %>%
```

```
select(Entity, Year, Continent, GDP_per_capita_PPP, GDP_Growth)

# Remove rows with missing GDP growth values (e.g., the first year for each country)
gdp_growth <- final_data %>%
  filter(!is.na(GDP_Growth)) # Exclude rows where GDP_Growth couldn't be calculated
head(gdp_growth)
```

```
## # A tibble: 6 x 5
## # Groups:   Entity [1]
##   Entity      Year Continent GDP_per_capita_PPP GDP_Growth
##   <chr>      <dbl> <chr>          <dbl>         <dbl>
## 1 Afghanistan 2003 Asia          1292.         0.927
## 2 Afghanistan 2004 Asia          1260.        -2.50
## 3 Afghanistan 2005 Asia          1352.         7.32
## 4 Afghanistan 2006 Asia          1367.         1.08
## 5 Afghanistan 2007 Asia          1528.        11.8
## 6 Afghanistan 2008 Asia          1557.         1.86
```

```
# Calculate the average GDP growth for each continent by year
continent_growth <- gdp_growth %>%
  group_by(Continent, Year) %>% # Group data by continent and year
  summarise(
    avg_growth = mean(GDP_Growth, na.rm = TRUE) # Calculate average GDP growth, ignoring missing values
  )
```

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

```
head(continent_growth)
```

```
## # A tibble: 6 x 3
## # Groups:   Continent [1]
##   Continent      Year avg_growth
##   <chr>      <dbl>     <dbl>
## 1 Africa      1991     -0.316
## 2 Africa      1992     -1.31
## 3 Africa      1993     -1.40
## 4 Africa      1994     -0.843
## 5 Africa      1995      2.99
## 6 Africa      1996      3.89
```

```
# Plot average GDP growth over time by continent
ggplot(continent_growth, aes(x = Year, y = avg_growth, color = Continent)) +
  geom_line(size = 1) + # Draw lines for each continent

# Highlight regions where growth exceeds the 7% target
geom_rect(
  data = continent_growth %>% filter(avg_growth > 7), # Filter data for values >7%
  aes(xmin = Year, xmax = Year, ymin = 7, ymax = avg_growth, fill = Continent),
  alpha = 0.2, inherit.aes = FALSE
) +
```

```

# Add a dashed red line to indicate the 7% target
geom_hline(yintercept = 7, linetype = "dashed", color = "red", size = 0.8) +
annotate("text", x = 2015, y = 7.5,
        label = "7% Target", color = "red") +

# Mark significant events with vertical lines and labels
geom_vline(xintercept = 2008, linetype = "dashed", color = "black", size = 0.8) +
geom_vline(xintercept = 2020, linetype = "dashed", color = "black", size = 0.8) +
annotate("text", x = 2008, y = -8,
        label = "2008 Crisis", color = "black", angle = 90, vjust = -0.5) +
annotate("text", x = 2020, y = -8,
        label = "COVID-19", color = "black", angle = 90, vjust = -0.5) +

# Add titles and axis labels
labs(
  title = "Average GDP Growth Over Time by Continent",
  x = "Year",
  y = "Average GDP Growth (%)",
  color = "Continent",
  fill = "Target Met"
) +

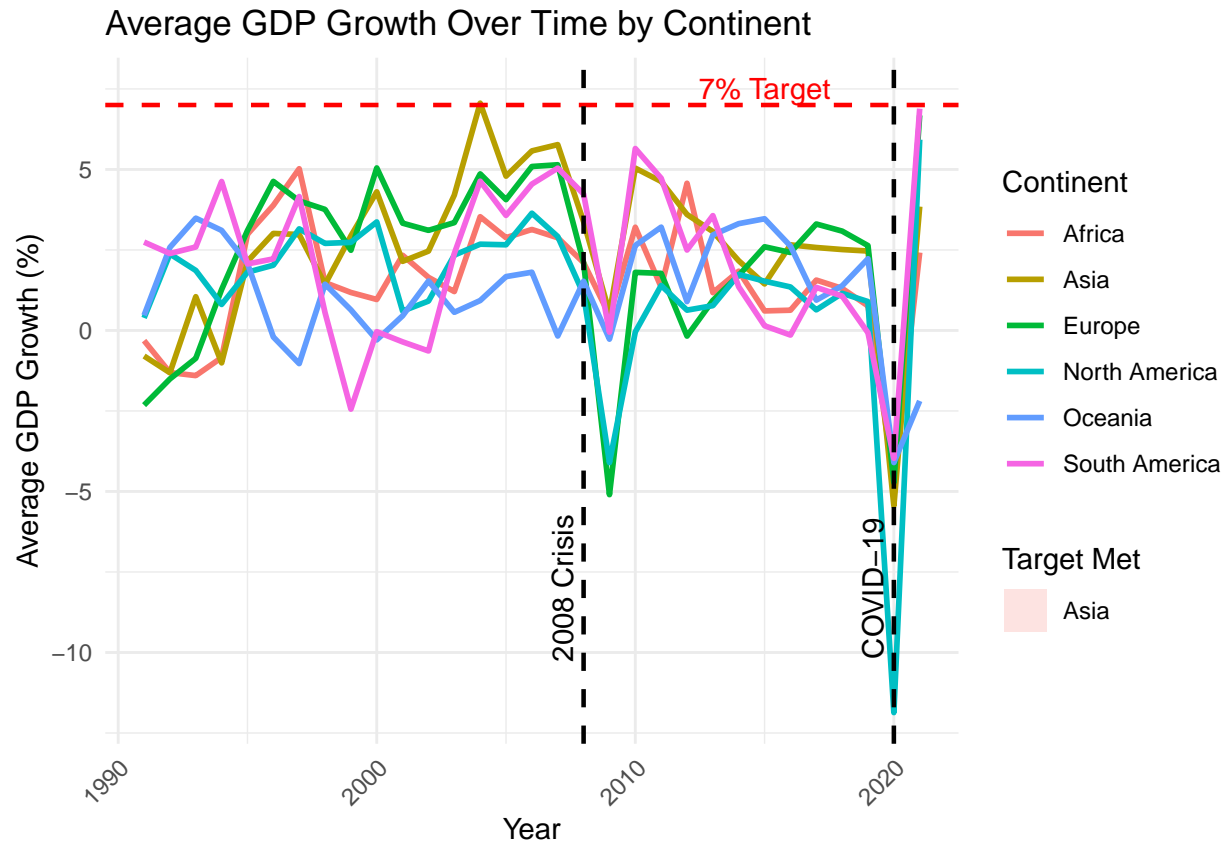
# Apply a clean and simple theme
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for readability

```

```

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```



```
# Step 1: Filter data for 2016 onwards and calculate average GDP growth by country
average_growth <- gdp_growth %>%
  filter(Year >= 2016) %>% # Only include data from 2016 onwards
  group_by(Entity) %>% # Group by country
  summarise(
    AverageGrowth = mean(GDP_Growth, na.rm = TRUE) # Calculate the mean GDP growth, ignoring missing values
  )

# Step 2: Load world map data and standardize country names
world_map <- map_data("world") # Load built-in world map data
world_map <- world_map %>%
  mutate(region = case_when(
    region == "USA" ~ "United States", # Standardize USA to match the dataset
    region == "UK" ~ "United Kingdom", # Standardize UK to match the dataset
    region == "Greenland" ~ "Denmark", # Assign Greenland's GDP to Denmark
    TRUE ~ region # Leave other names unchanged
  ))

# Step 3: Merge the map data with the average GDP growth data
merged_data <- world_map %>%
  left_join(average_growth, by = c("region" = "Entity")) # Match map regions with country names

# Step 4: Plot the map
ggplot(data = merged_data) +
  geom_polygon(
    aes(x = long, y = lat, group = group, fill = AverageGrowth), # Fill polygons by average growth
  )
```

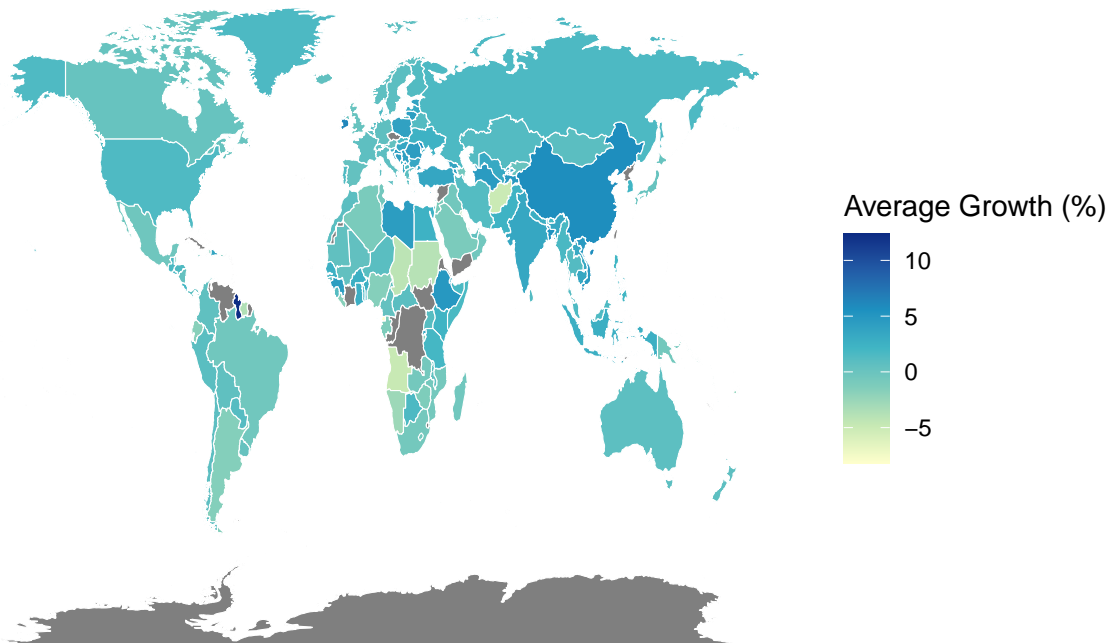
```

    color = "white", size = 0.2 # Add white borders between countries
  ) +
  scale_fill_distiller(
    palette = "YlGnBu", # Yellow-Green-Blue color palette for the gradient
    direction = 1, # Gradient goes from low to high values
    name = "Average Growth (%)" # Legend title
  ) +
  labs(
    title = "Average GDP Growth (2016 Onwards)", # Plot title
    subtitle = "Coloured by Average Growth Rate", # Plot subtitle
    x = "", # Remove x-axis label
    y = "" # Remove y-axis label
  ) +
  theme_minimal() +
  theme(
    axis.text = element_blank(), # Hide axis labels
    axis.ticks = element_blank(), # Remove axis ticks
    panel.grid = element_blank(), # Remove grid lines
    plot.title = element_text(hjust = 0.5, face = "bold"), # Centered bold title
    plot.subtitle = element_text(hjust = 0.5) # Centered subtitle
  )

```

Average GDP Growth (2016 Onwards)

Coloured by Average Growth Rate



```

# Plot the density distribution of GDP growth across continents
ggplot(final_data, aes(x = GDP_Growth, fill = Continent)) +
  geom_density(alpha = 0.7) + # Create smoothed density curves with transparency

```

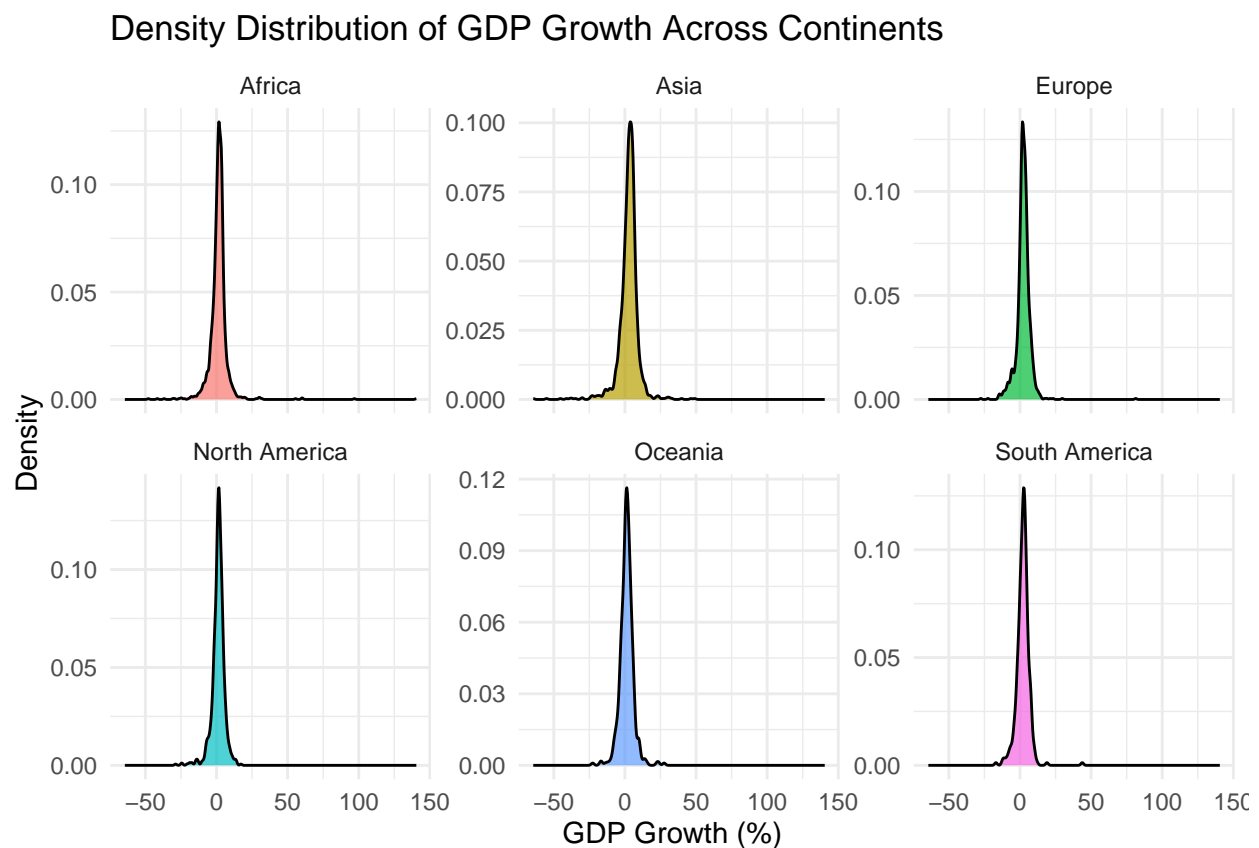
```

facet_wrap(~ Continent, scales = "free_y") + # Create separate plots for each continent, with indepe
labs(
  title = "Density Distribution of GDP Growth Across Continents", # Plot title
  x = "GDP Growth (%)", # X-axis label
  y = "Density" # Y-axis label
) +
theme_minimal() + # Apply a clean theme
theme(
  legend.position = "none" # Remove the legend since it's redundant with facets
)

```

```
## Warning: Removed 194 rows containing non-finite outside the scale range
```

```
## ('stat_density()').
```



```

# Function to identify outliers in GDP growth data
identify_outliers <- function(data) {
  data %>%
    group_by(Year) %>% # Group data by year for outlier detection
    mutate(
      Q1 = quantile(GDP_Growth, 0.25, na.rm = TRUE), # Calculate the 1st quartile (25th percentile)
      Q3 = quantile(GDP_Growth, 0.75, na.rm = TRUE), # Calculate the 3rd quartile (75th percentile)
      IQR = Q3 - Q1, # Compute the interquartile range (IQR)
      is_outlier = GDP_Growth < (Q1 - 1.5 * IQR) | GDP_Growth > (Q3 + 1.5 * IQR)
      # Flag rows where GDP_Growth is below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR
    ) %>%

```

```

    filter(is_outlier) # Keep only rows flagged as outliers
}

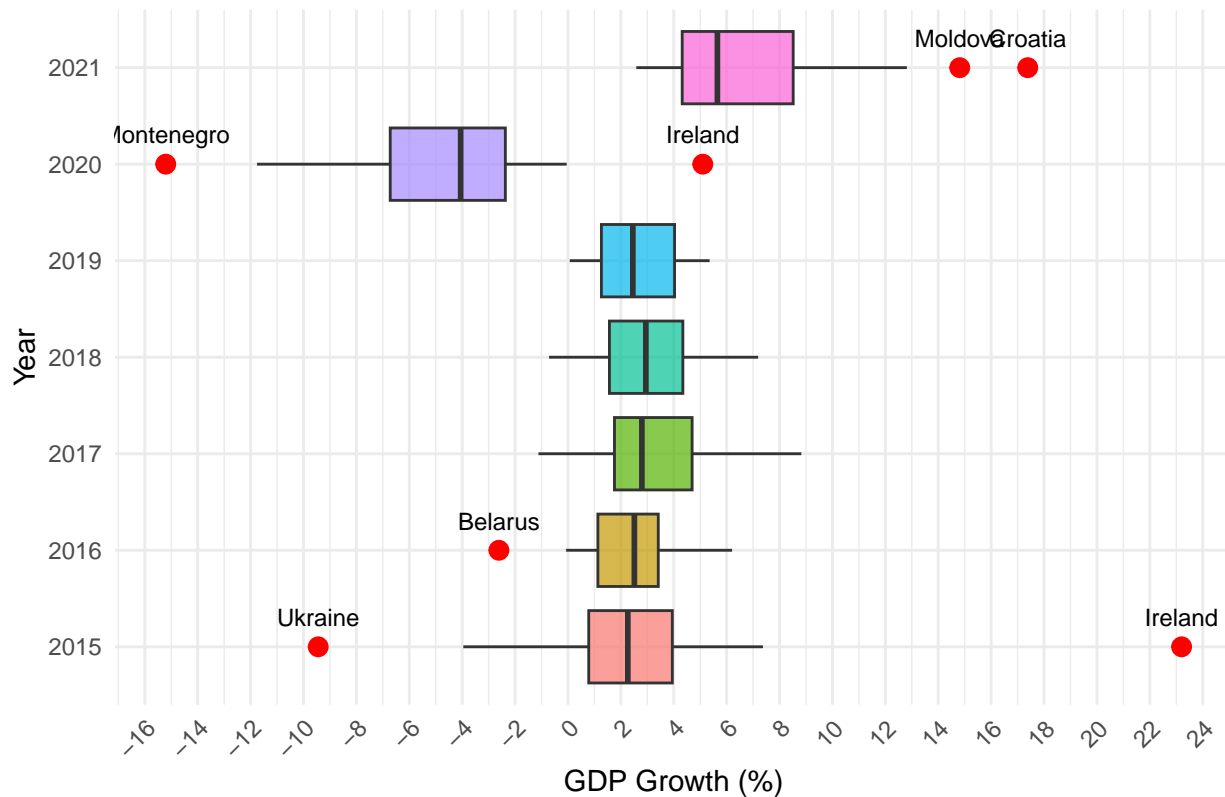
# Step 1: Filter data for Europe from 2015 onwards
europe_data <- gdp_growth %>%
  filter(Continent == "Europe" & Year >= 2015) # Focus on Europe and data from 2015 onwards

# Step 2: Identify outliers for Europe using the identify_outliers function
europe_outliers <- identify_outliers(europe_data)

# Step 3: Create a boxplot for GDP growth in Europe, highlighting and labeling outliers
ggplot(europe_data, aes(x = GDP_Growth, y = as.factor(Year), fill = as.factor(Year))) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) + # Boxplot without default outlier points
  geom_point(
    data = europe_outliers, aes(x = GDP_Growth, y = as.factor(Year)),
    color = "red", size = 3 # Add red points for outliers
  ) +
  geom_text(
    data = europe_outliers, aes(x = GDP_Growth, y = as.factor(Year), label = Entity),
    vjust = -1.2, size = 3, color = "black" # Label outliers above red dots
  ) +
  labs(
    title = "Box Plot of GDP Growth Rate (%) in Europe",
    x = "GDP Growth (%)", # Label for x-axis
    y = "Year", # Label for y-axis
    fill = "Year" # Legend title
  ) +
  theme_minimal() + # Apply a clean theme
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for better readability
    legend.position = "none" # Remove legend
  ) +
  scale_x_continuous(
    breaks = seq(
      floor(min(europe_data$GDP_Growth, na.rm = TRUE)),
      ceiling(max(europe_data$GDP_Growth, na.rm = TRUE)), by = 2 # Adjust x-axis ticks
    )
  )
)

```

Box Plot of GDP Growth Rate (%) in Europe



```
# Step 1: Filter data for North America from 2015 onwards
north_america_data <- gdp_growth %>%
  filter(Continent == "North America" & Year >= 2015) # Focus on North America and data from 2015 onwards

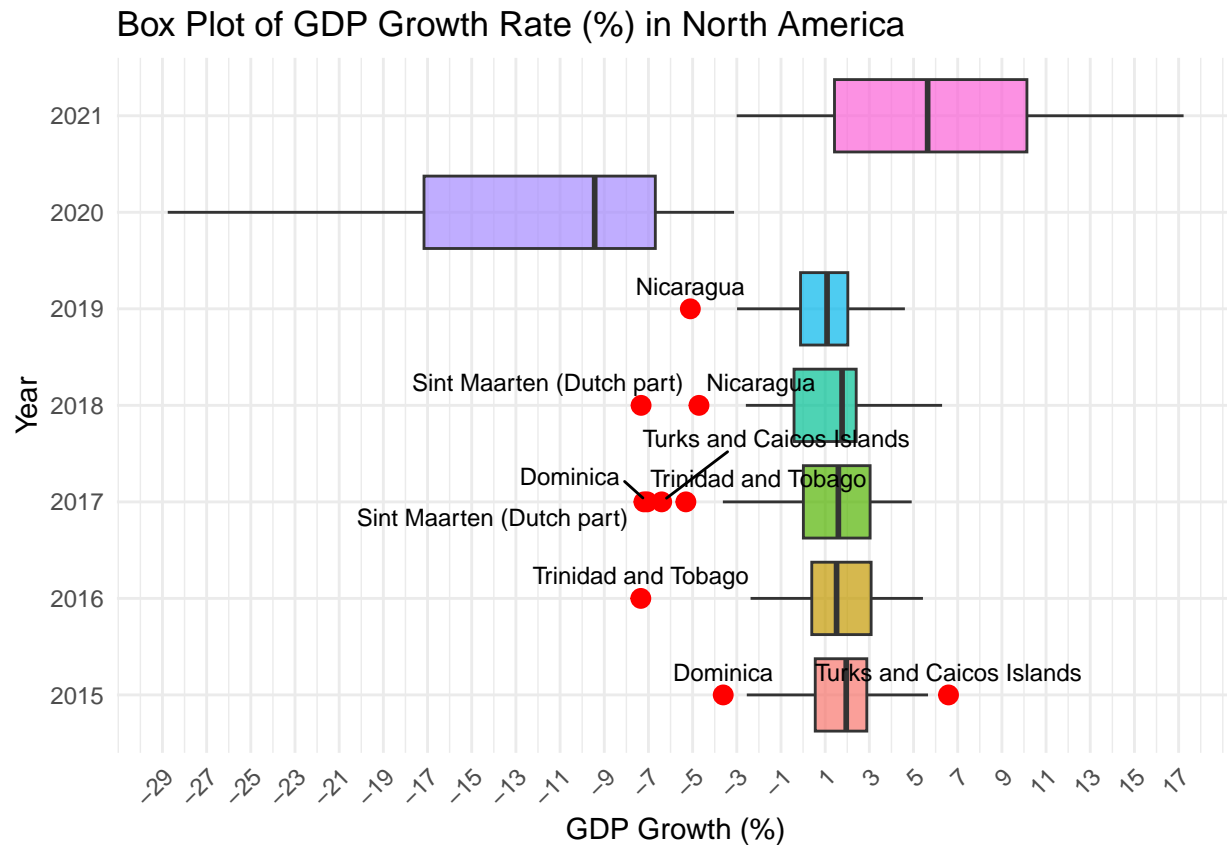
# Step 2: Identify outliers for North America using the identify_outliers function
north_america_outliers <- identify_outliers(north_america_data)

# Step 3: Create a boxplot for GDP growth in North America, highlighting and labeling outliers
ggplot(north_america_data, aes(x = GDP_Growth, y = as.factor(Year), fill = as.factor(Year))) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) + # Boxplot without default outlier points
  geom_point(
    data = north_america_outliers, aes(x = GDP_Growth, y = as.factor(Year)),
    color = "red", size = 3 # Add red points for outliers
  ) +
  geom_text_repel(
    data = north_america_outliers, aes(x = GDP_Growth, y = as.factor(Year), label = Entity),
    size = 3, color = "black", nudge_y = 0.2, box.padding = 0.3, point.padding = 0.2 # Smart positioning
  ) +
  labs(
    title = "Box Plot of GDP Growth Rate (%) in North America",
    x = "GDP Growth (%)", # Label for x-axis
    y = "Year", # Label for y-axis
    fill = "Year" # Legend title
  ) +
  theme_minimal() + # Apply a clean theme
  theme(
```

```

axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for better readability
legend.position = "none" # Remove legend
) +
scale_x_continuous(
  breaks = seq(
    floor(min(north_america_data$GDP_Growth, na.rm = TRUE)),
    ceiling(max(north_america_data$GDP_Growth, na.rm = TRUE)), by = 2 # Adjust x-axis ticks
  )
)

```



```

# LDCs list
ldc_list <- c(
  "Angola", "Benin", "Burkina Faso", "Burundi", "Chad", "Comoros", "DR Congo", "Djibouti",
  "Eritrea", "Ethiopia", "Gambia", "Guinea", "Guinea-Bissau", "Lesotho", "Liberia",
  "Madagascar", "Malawi", "Mali", "Mauritania", "Mozambique", "Niger", "Rwanda",
  "Senegal", "Sierra Leone", "Somalia", "South Sudan", "Sudan", "Tanzania",
  "Togo", "Uganda", "Zambia", "Afghanistan", "Bangladesh", "Bhutan", "Cambodia",
  "Laos", "Myanmar", "Nepal", "Timor-Leste", "Yemen", "Kiribati", "Solomon Islands",
  "Tuvalu", "Vanuatu", "Haiti"
)

# MDCs list
mdc_list <- c(
  "Austria", "Belgium", "Denmark", "Finland", "France", "Germany", "Italy",
  "Netherlands", "Norway", "Sweden", "Switzerland", "United Kingdom",

```

```

"Canada", "United States", "Japan", "South Korea", "Singapore",
"Australia", "New Zealand"
)

#Filter
ldc_data <- gdp_growth %>%
  filter(Entity %in% ldc_list)

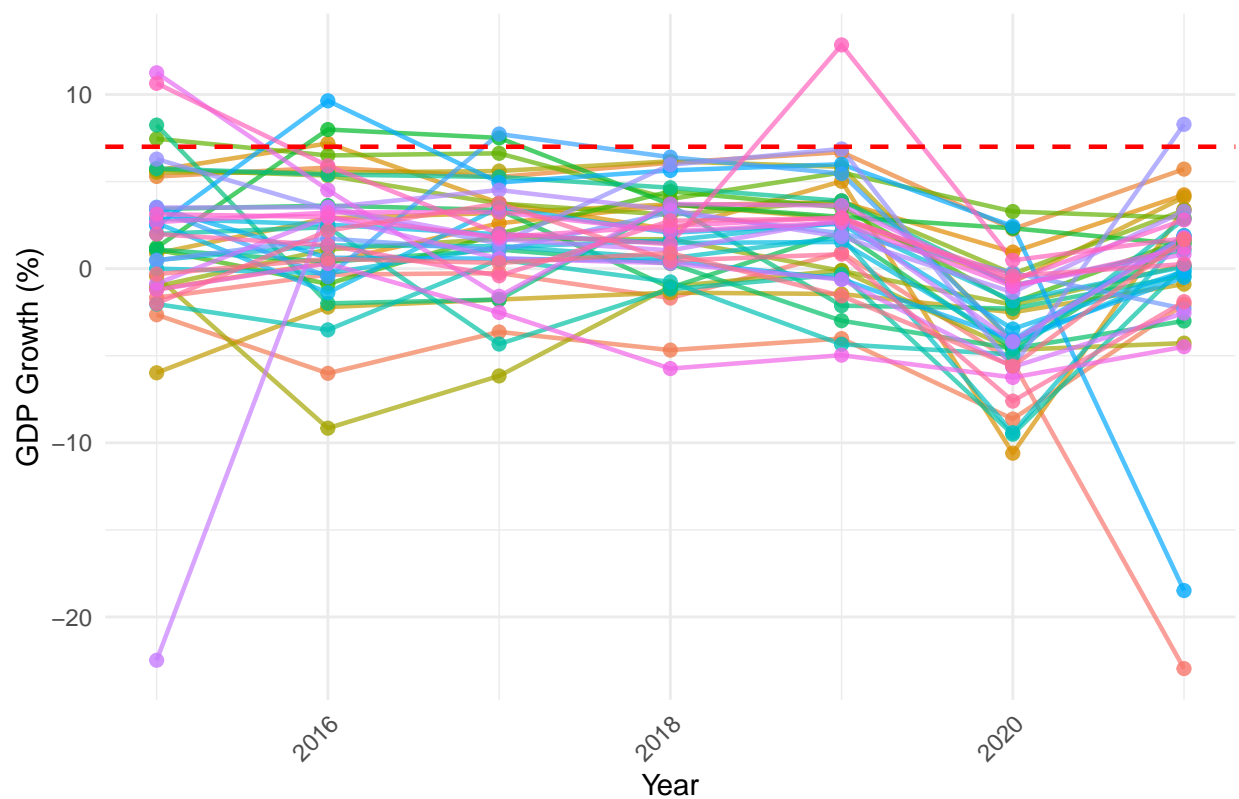
# Filter data for MDCs
mdc_data <- gdp_growth %>%
  filter(Entity %in% mdc_list)

# Step 1: Filter data for LDCs from 2015 onwards
ldc_growth <- gdp_growth %>%
  filter(Entity %in% ldc_list & Year >= 2015) # Focus on LDCs and data from 2015 onwards

# Step 2: Create a line plot for GDP growth in LDCs
ggplot(ldc_growth, aes(x = Year, y = GDP_Growth, color = Entity, group = Entity)) +
  geom_line(size = 0.8, alpha = 0.7) + # Add lines for each country with slight transparency
  geom_point(size = 2, alpha = 0.8) + # Add points for each year's GDP growth
  geom_hline(yintercept = 7, linetype = "dashed", color = "red", size = 0.8) + # Highlight the 7% target
  labs(
    title = "Economic Growth for Least Developed Countries (LDCs)", # Plot title
    x = "Year", # X-axis label
    y = "GDP Growth (%)", # Y-axis label
    color = "Country" # Legend title
  ) +
  theme_minimal() + # Apply a clean theme
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for better readability
    legend.position = "none" # Remove legend for simplicity
  )

```

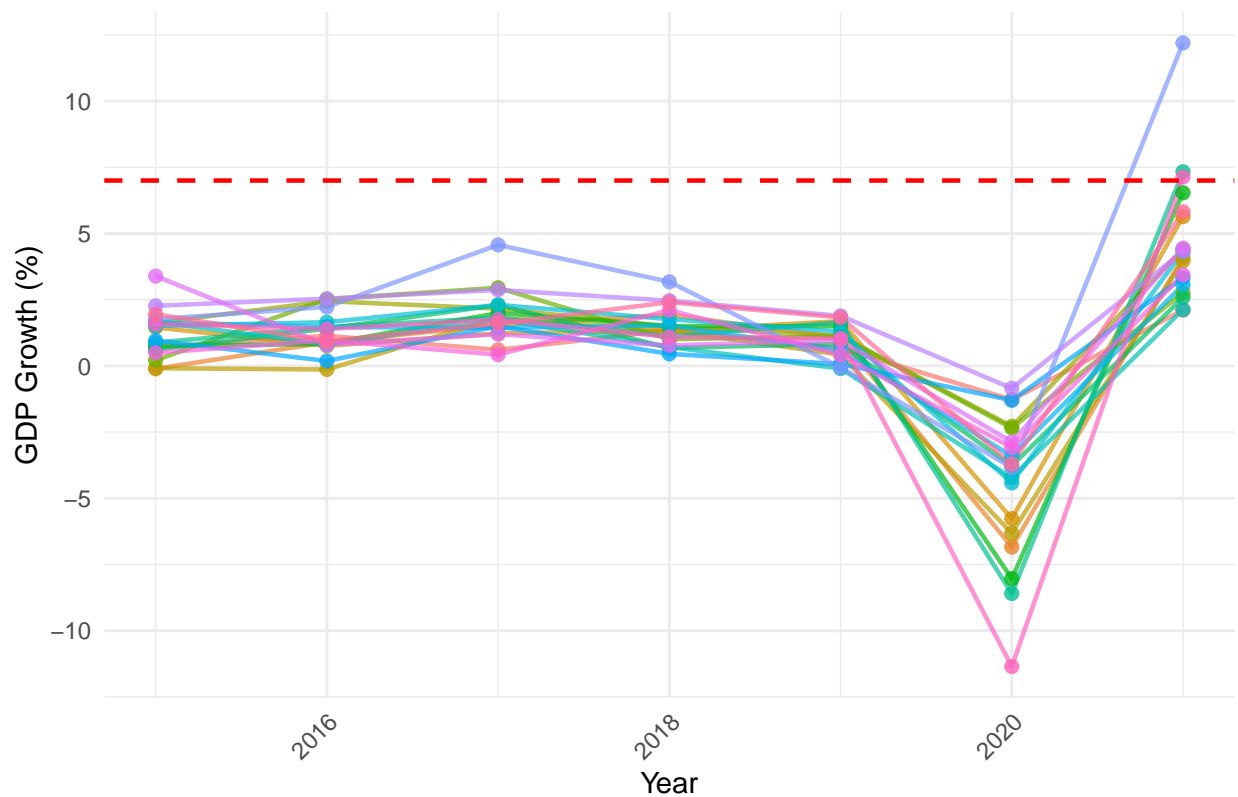
Economic Growth for Least Developed Countries (LDCs)



```
# Step 1: Filter data for MDCs from 2015 onwards
mdc_growth <- gdp_growth %>%
  filter(Entity %in% mdc_list & Year >= 2015) # Focus on MDCs and data from 2015 onwards

# Step 2: Create a line plot for GDP growth in MDCs
ggplot(mdc_growth, aes(x = Year, y = GDP_Growth, color = Entity, group = Entity)) +
  geom_line(size = 0.8, alpha = 0.7) + # Add lines for each country with slight transparency
  geom_point(size = 2, alpha = 0.8) + # Add points for each year's GDP growth
  geom_hline(yintercept = 7, linetype = "dashed", color = "red", size = 0.8) + # Highlight the 7% target
  labs(
    title = "Economic Growth for More Developed Countries (MDCs)", # Plot title
    x = "Year", # X-axis label
    y = "GDP Growth (%)", # Y-axis label
    color = "Country" # Legend title
  ) +
  theme_minimal() + # Apply a clean theme
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for better readability
    legend.position = "none" # Remove legend for simplicity
  )
```

Economic Growth for More Developed Countries (MDCs)



```
# Step 1: Filter and classify data for Asia
asia_data <- gdp_growth %>%
  filter(Continent == "Asia" & Year >= 2015) %>% # Include Asian countries from 2015 onwards
  mutate(
    Development_Status = ifelse(Entity %in% ldc_list, "LDC", "MDC") # Classify countries as LDC or MDC
  )

asia_growth_comparison <- asia_data %>%
  group_by(Year, Development_Status) %>% # Group by year and development status
  summarise(Average_GDP_Growth = mean(GDP_Growth, na.rm = TRUE)) # Calculate the average GDP growth
```

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

```
# Step 2: Bar chart comparing LDCs and MDCs in Asia
ggplot(asia_growth_comparison, aes(x = as.factor(Year), y = Average_GDP_Growth, fill = Development_Status)) +
  geom_bar(
    stat = "identity", position = position_dodge(width = 0.8), # Dodged bar chart
    alpha = 0.9, color = "black" # Slight transparency with black borders
  ) +
  geom_text(
    aes(label = round(Average_GDP_Growth, 1)), # Add value labels to bars
    position = position_dodge(width = 0.8),
    vjust = -0.5, size = 3, color = "black"
  ) +
```

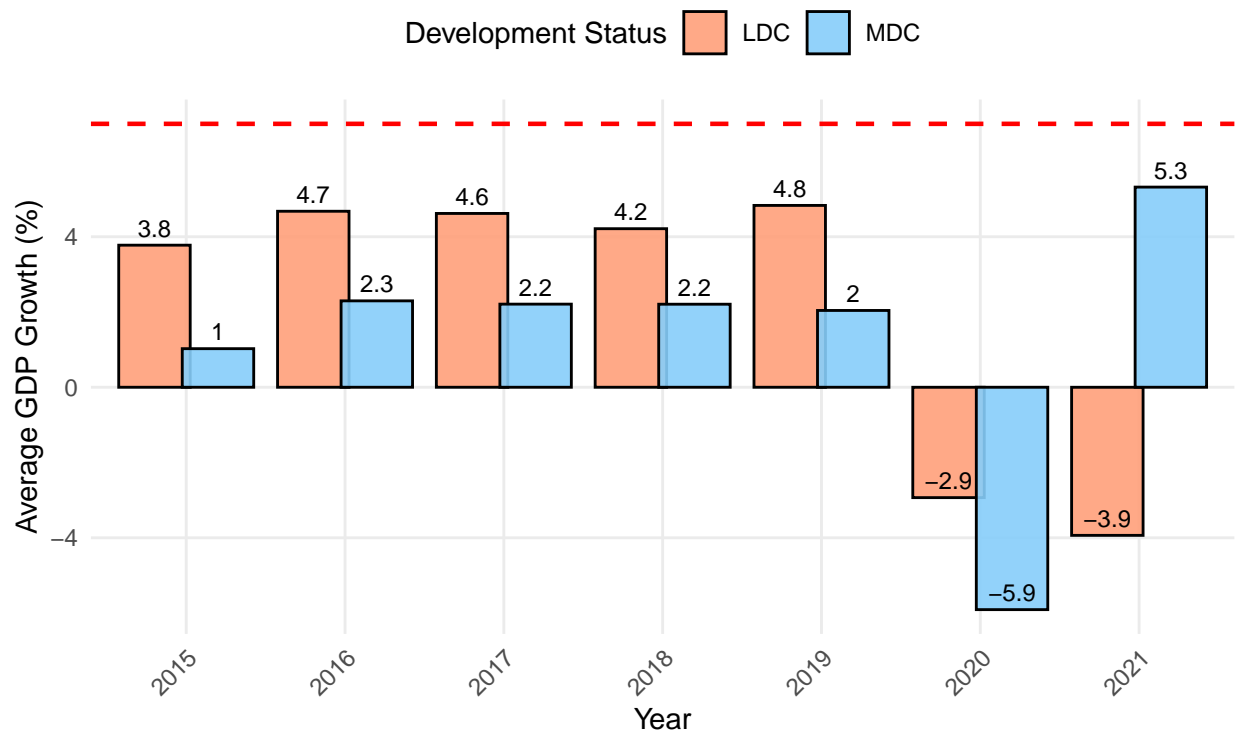
```

geom_hline(yintercept = 7, linetype = "dashed", color = "red", size = 0.8) + # Dashed red line for target
labs(
  title = "Economic Growth Comparison: LDCs vs MDCs in Asia", # Plot title
  subtitle = "Average GDP Growth (%) per Year", # Plot subtitle
  x = "Year", # X-axis label
  y = "Average GDP Growth (%)", # Y-axis label
  fill = "Development Status" # Legend title
) +
scale_fill_manual(values = c("LDC" = "#FFA07A", "MDC" = "#87CEFA")) + # Custom colors for LDC and MDC
theme_minimal() + # Apply a clean theme
theme(
  axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for readability
  plot.title = element_text(face = "bold", size = 14), # Bold title with adjusted size
  plot.subtitle = element_text(size = 12), # Subtitle with adjusted size
  legend.position = "top", # Move legend to the top
  panel.grid.minor = element_blank() # Remove minor grid lines for a cleaner look
)

```

Economic Growth Comparison: LDCs vs MDCs in Asia

Average GDP Growth (%) per Year



```

# Step 1: Filter data for countries exceeding the 7% GDP growth target
target_data <- gdp_growth %>%
  filter(GDP_Growth > 7) %>% # Include only rows where GDP growth exceeds 7%
  group_by(Continent, Year) %>% # Group by continent and year
  summarise(
    Number_of_Countries = n_distinct(Entity) # Count the number of distinct countries
  )

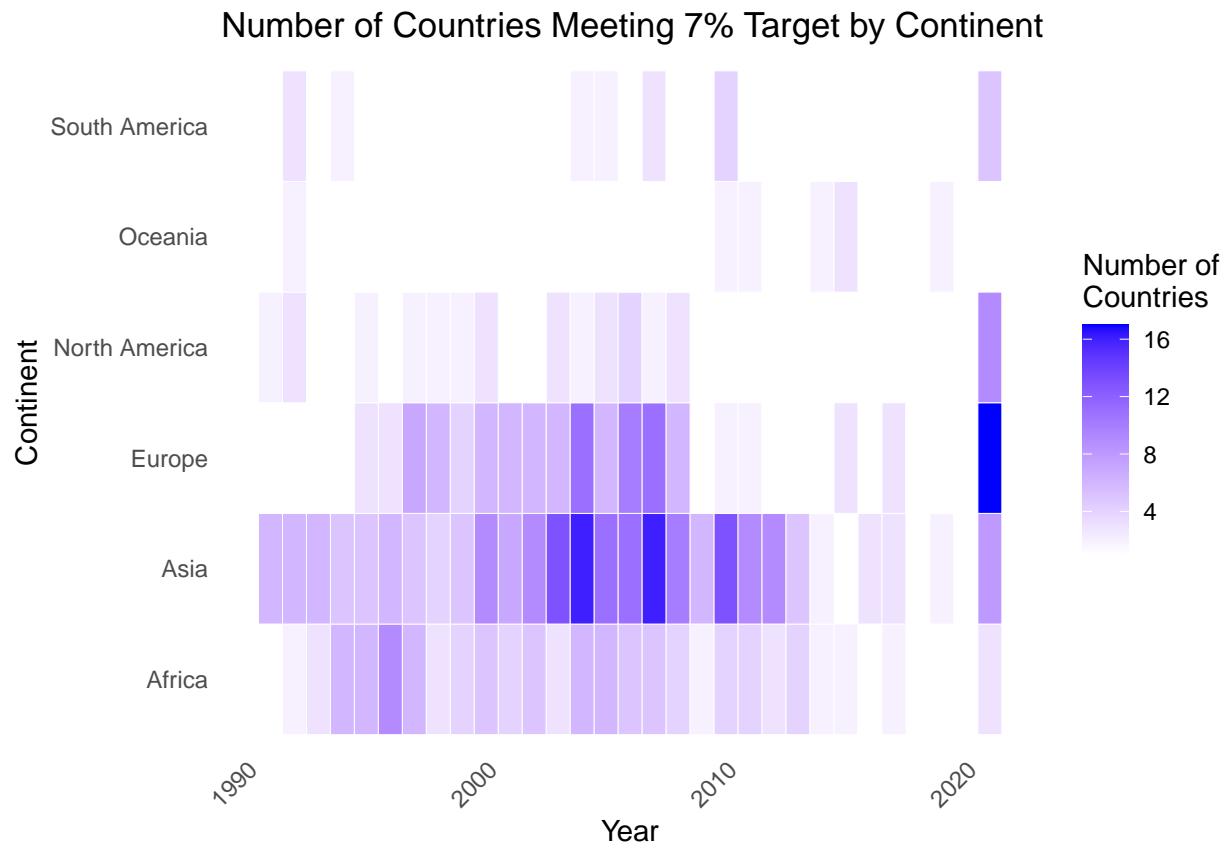
```

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

```
head(target_data)
```

```
## # A tibble: 6 x 3
## # Groups:   Continent [1]
##   Continent Year Number_of_Countries
##   <chr>      <dbl>          <int>
## 1 Africa    1991              1
## 2 Africa    1992              2
## 3 Africa    1993              3
## 4 Africa    1994              6
## 5 Africa    1995              6
## 6 Africa    1996              9
```

```
# Step 2: Create a heatmap for the number of countries meeting the target
ggplot(target_data, aes(x = Year, y = Continent, fill = Number_of_Countries)) +
  geom_tile(color = "white") + # Heatmap with white borders between tiles
  scale_fill_gradient(
    low = "white", high = "blue", na.value = "grey50", # Gradient from white to blue
    name = "Number of\nCountries" # Legend title
  ) +
  labs(
    title = "Number of Countries Meeting 7% Target by Continent", # Plot title
    x = "Year", # X-axis label
    y = "Continent" # Y-axis label
  ) +
  theme_minimal() + # Apply a clean theme
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for readability
    panel.grid.major = element_blank(), # Remove major grid lines for a cleaner look
    panel.grid.minor = element_blank() # Remove minor grid lines
  )
```



##Target 2

```
# Step 1: Calculate NEET change over time for each country
neet_change <- neet_data %>%
  arrange(Code, Year) %>% # Sort by country code and year
  group_by(Code) %>%
  filter(!is.na(Share)) %>% # Remove rows with missing Share values
  summarize(
    NEET_Change = last(Share) - first(Share), # Calculate NEET change
    .groups = "drop"
  )
head(neet_change)
```

```
## # A tibble: 6 x 2
##   Code NEET_Change
##   <chr>      <dbl>
## 1 ""        -4.13
## 2 "ABW"         0
## 3 "AFG"        27.8
## 4 "AGO"       -9.27
## 5 "ALB"       -16.0
## 6 "ARE"         0
```

```
# Step 2: Load world shapefile
world <- rnaturalearth::ne_countries(scale = "medium", returnclass = "sf")
```

```

# Step 3: Separate Greenland from Denmark in the `world` dataset
world <- world %>%
  mutate(
    iso_a3 = case_when(
      name_long == "Greenland" ~ "GRL", # Assign Greenland's ISO code
      TRUE ~ iso_a3
    )
  )

# Step 4: Ensure Greenland is represented in NEET data
# If Greenland's data is missing, duplicate Denmark's NEET data for Greenland
if (!"GRL" %in% neet_change$Code) {
  greenland_data <- neet_change %>%
    filter(Code == "DNK") %>% # Use Denmark's data
    mutate(Code = "GRL") # Assign Greenland's code
  neet_change <- bind_rows(neet_change, greenland_data) # Add Greenland to NEET data
}

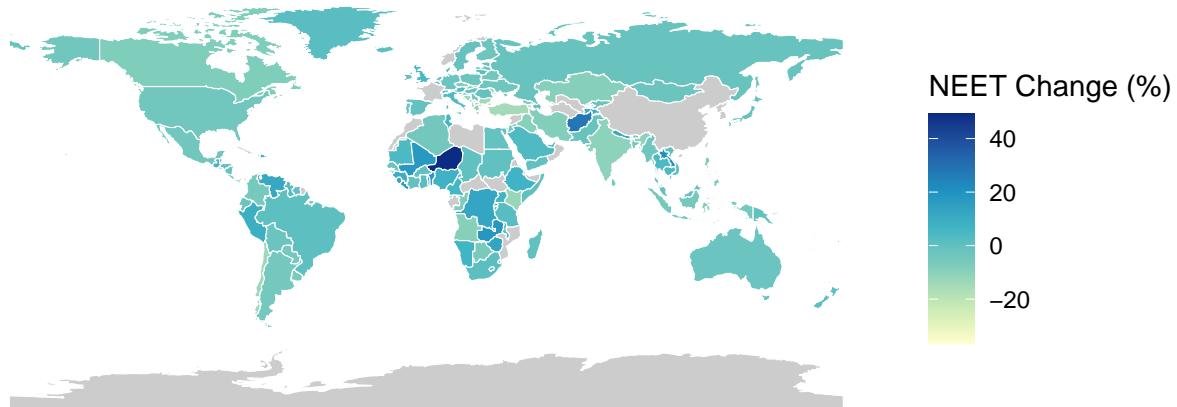
# Step 5: Merge NEET data with world shapefile
world_neet <- world %>%
  left_join(neet_change, by = c("iso_a3" = "Code"))

# Step 6: Plot the map
ggplot(world_neet) +
  geom_sf(aes(fill = NEET_Change), color = "white", size = 0.1) +
  scale_fill_distiller(
    palette = "YlGnBu",
    direction = 1,
    name = "NEET Change (%)",
    na.value = "grey80"
  ) +
  labs(
    title = "Change in NEET (% of Youth Population)",
    subtitle = "Progress in reducing NEET rates by country",
    x = NULL,
    y = NULL
  ) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    panel.grid = element_blank(),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5)
  )

```

Change in NEET (% of Youth Population)

Progress in reducing NEET rates by country



```
# Merge NEET data with continent mapping
merged_data <- neet_data %>%
  left_join(continents %>% select(Entity, Continent), by = "Entity") # Add continent information

# Filter data for the years 2016-2021 and remove missing values
filtered_data <- merged_data %>%
  filter(
    Year >= 2016 & Year <= 2021, # Focus on the years 2016-2021
    !is.na(Continent),           # Remove rows with missing continent data
    !is.na(Share)                # Remove rows with missing NEET proportions
  )

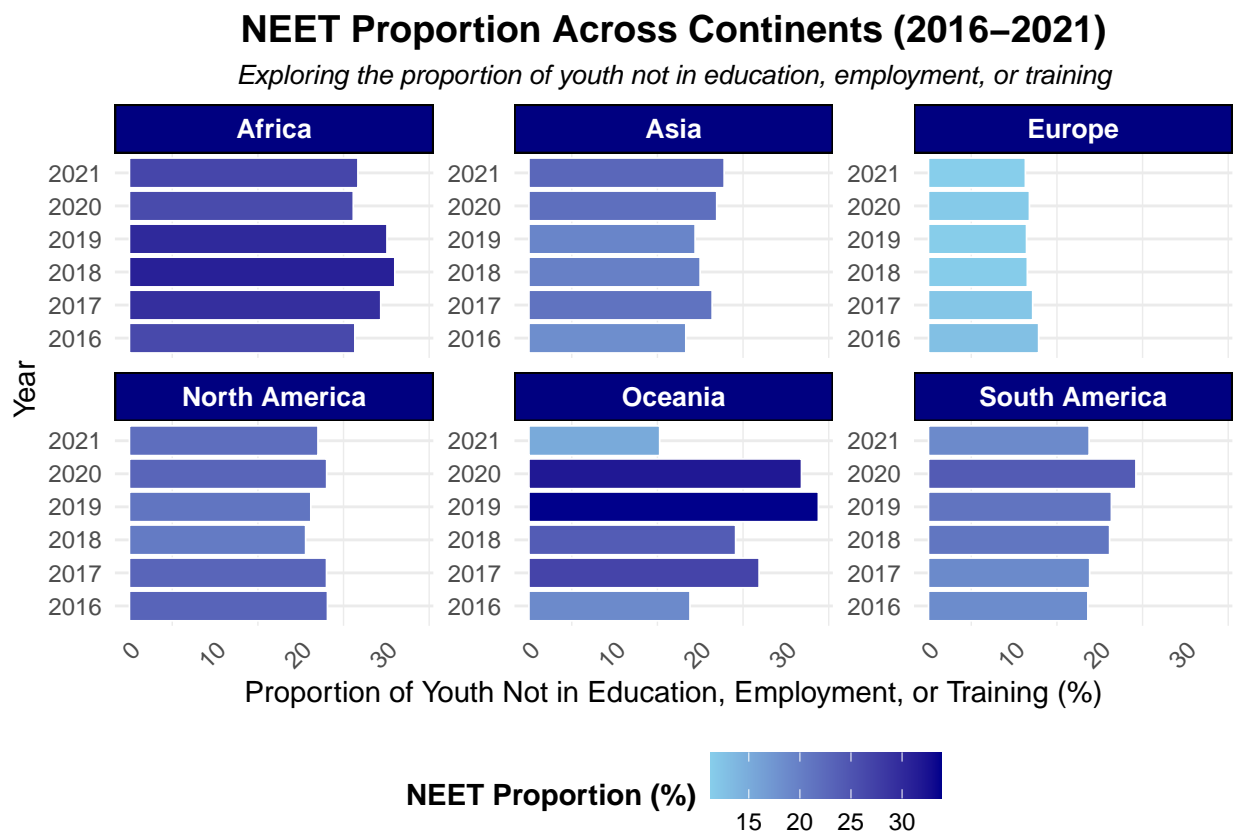
# Calculate the mean NEET proportion for each continent and year
aggregated_data <- filtered_data %>%
  group_by(Continent, Year) %>% # Group by continent and year
  summarise(
    mean_neet = mean(Share, na.rm = TRUE), # Calculate average NEET proportion
    .groups = "drop"
  )

# Create a horizontal bar chart with facets by continent
ggplot(aggregated_data, aes(x = mean_neet, y = as.factor(Year), fill = mean_neet)) +
  geom_bar(stat = "identity", position = "dodge", color = "white", size = 0.3) + # Horizontal bars with
  facet_wrap(~ Continent, scales = "free_y") + # Separate facets for each continent, free y-axis
  scale_fill_gradient(
    low = "skyblue", high = "darkblue", # Gradient from light to dark blue
  )
```

```

name = "NEET Proportion (%)" # Legend title
) +
labs(
  title = "NEET Proportion Across Continents (2016-2021)", # Plot title
  subtitle = "Exploring the proportion of youth not in education, employment, or training", # Subtit
  x = "Proportion of Youth Not in Education, Employment, or Training (%)", # X-axis label
  y = "Year" # Y-axis label
) +
theme_minimal() + # Apply a clean theme
theme(
  legend.position = "bottom", # Place legend at the bottom
  legend.title = element_text(face = "bold"), # Bold legend title
  legend.text = element_text(size = 9), # Adjust legend text size
  strip.background = element_rect(fill = "navy"), # Dark background for facet labels
  strip.text = element_text(color = "white", face = "bold", size = 10), # Bold and white facet text
  plot.title = element_text(hjust = 0.5, face = "bold", size = 14), # Centered and bold plot title
  plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 10), # Centered italic subtitle
  axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for clarity
  panel.grid.major.x = element_blank() # Remove vertical grid lines
)

```



```

# Step 1: Merge NEET data with continent mapping
merged_data <- neet_data %>%
  left_join(continents %>% select(Entity, Continent), by = "Entity") # Add continent information

```

```

# Step 2: Filter data for the years 2016-2021 and remove missing values
filtered_data <- merged_data %>%
  filter(
    Year >= 2016 & Year <= 2021, # Focus on the years 2016-2021
    !is.na(Continent),           # Remove rows with missing continent data
    !is.na(Share)                # Remove rows with missing NEET proportions
  )

# Step 3: Identify countries with complete data for all years (2016-2021)
complete_countries <- filtered_data %>%
  group_by(Entity) %>%
  summarise(year_count = n_distinct(Year)) %>%
  filter(year_count == 6) %>% # Keep only countries with data for all six years
  pull(Entity)

# Step 4: Filter dataset for these complete countries
complete_data <- filtered_data %>%
  filter(Entity %in% complete_countries)

# Step 5: Calculate mean NEET proportion by continent and year
continent_data <- complete_data %>%
  group_by(Continent, Year) %>%
  summarise(mean_neet = mean(Share, na.rm = TRUE), .groups = "drop") # Calculate average NEET proportion

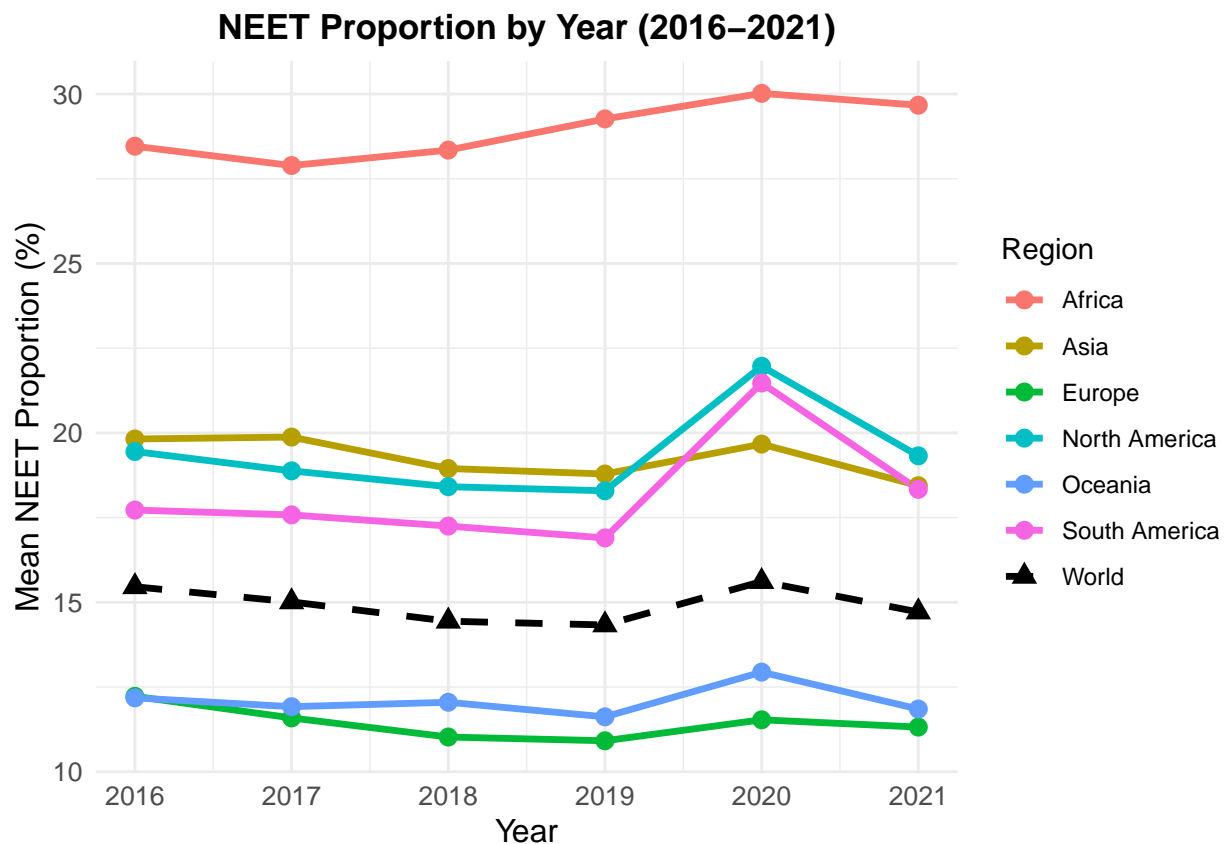
# Step 6: Calculate global average NEET proportion by year
global_data <- complete_data %>%
  group_by(Year) %>%
  summarise(mean_neet = mean(Share, na.rm = TRUE), .groups = "drop") %>%
  mutate(Continent = "World") # Label as "World"

# Step 7: Combine continent-level and global-level data
final_data <- bind_rows(continent_data, global_data)

# Step 8: Create the line graph
ggplot(final_data, aes(x = Year, y = mean_neet, color = Continent, linetype = Continent)) +
  geom_line(size = 1.2) + # Add lines for each region
  geom_point(size = 3, aes(shape = Continent)) + # Add points for each year
  scale_color_manual(
    values = c(scales::hue_pal()(length(unique(continent_data$Continent))), "black"), # Unique colors
    name = "Region"
  ) +
  scale_linetype_manual(
    values = c(rep("solid", length(unique(continent_data$Continent))), "dashed"), # Solid for continents
    name = "Region"
  ) +
  scale_shape_manual(
    values = c(rep(16, length(unique(continent_data$Continent))), 17), # Dots for continents, triangle for world
    name = "Region"
  ) +
  labs(
    title = "NEET Proportion by Year (2016-2021)", # Plot title
    x = "Year", # X-axis label
    y = "Mean NEET Proportion (%)" # Y-axis label
  )

```

```
) +
theme_minimal() + # Apply a clean theme
theme(
  legend.position = "right", # Position legend to the right
  plot.title = element_text(hjust = 0.5, face = "bold"), # Center and bold the title
  axis.text = element_text(size = 10), # Adjust axis text size
  axis.title = element_text(size = 12) # Adjust axis title size
)
```



```
# Step 1: Merge NEET data with continent mapping
merged_data <- neet_data %>%
  left_join(continents %>% select(Entity, Continent), by = "Entity") # Add continent information

# Step 2: Filter data for 2016-2021 and remove missing values
filtered_data <- merged_data %>%
  filter(
    Year >= 2016 & Year <= 2021, # Focus on the years 2016-2021
    !is.na(Continent),             # Remove rows with missing continent data
    !is.na(Share)                  # Remove rows with missing NEET proportions
  )

# Step 3: Identify countries with complete data for all years
complete_countries <- filtered_data %>%
  group_by(Entity) %>%
  summarise(year_count = n_distinct(Year)) %>%
```

```

filter(year_count == 6) %>% # Retain countries with data for all six years
pull(Entity)

# Step 4: Filter dataset for these complete countries
complete_data <- filtered_data %>%
  filter(Entity %in% complete_countries)

# Step 5: Calculate total NEET population for each continent and year
continent_totals <- complete_data %>%
  group_by(Continent, Year) %>%
  summarise(total_neet = sum(Share, na.rm = TRUE), .groups = "drop") # Sum NEET proportions per continent

# Step 6: Index NEET population to 2016 for each continent
indexed_data <- continent_totals %>%
  group_by(Continent) %>%
  mutate(index = (total_neet / total_neet[Year == 2016]) * 100) %>% # Index to 2016
  ungroup()

# Step 7: Create the indexed growth line graph
ggplot(indexed_data, aes(x = Year, y = index, color = Continent, group = Continent)) +
  geom_line(size = 1.5, alpha = 0.8) + # Thick lines with slight transparency
  geom_point(size = 4, shape = 21, fill = "white", stroke = 1) + # Highlight points with white fill
  scale_color_brewer(palette = "Set1", name = "Continent") + # Use a colorblind-friendly palette

labs(
  title = "Growth of Total NEET Population (2016 = Base Year)", # Plot title
  subtitle = "Indexed NEET population changes across continents from 2016 onward", # Subtitle
  x = "Year", # X-axis label
  y = "Index (2016 = Base Year)" # Y-axis label
) +
theme_minimal() + # Apply a clean, modern theme
theme(
  legend.position = "right", # Position legend on the right
  legend.title = element_text(face = "bold", size = 12), # Bold legend title
  legend.text = element_text(size = 10), # Adjust legend text size
  plot.title = element_text(hjust = 0.5, face = "bold", size = 16), # Center and bold title
  plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 12), # Center subtitle
  axis.text.x = element_text(angle = 45, hjust = 1, size = 10), # Rotate x-axis labels for clarity
  axis.text.y = element_text(size = 10), # Adjust y-axis text size
  axis.title = element_text(size = 12, face = "bold"), # Bold axis titles
  panel.grid.major = element_line(color = "gray80", size = 0.5), # Subtle major grid lines
  panel.grid.minor = element_blank() # Remove minor grid lines
)

```

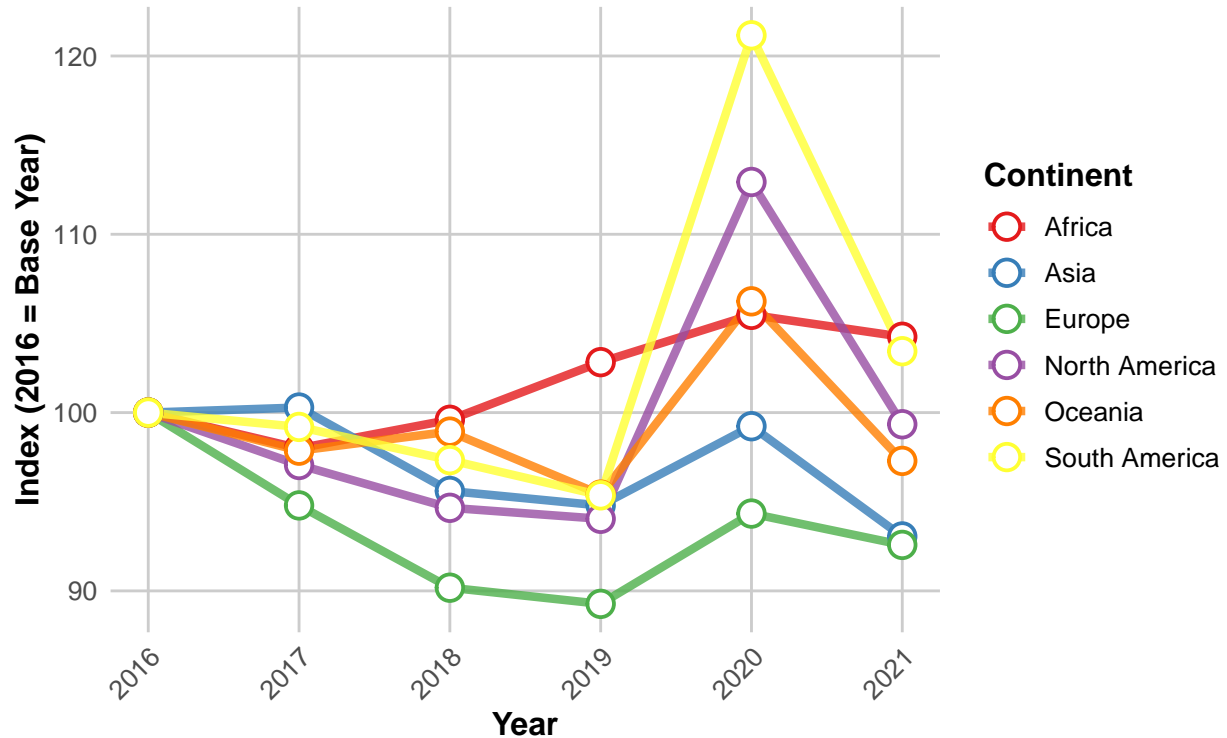
```

## Warning: The 'size' argument of 'element_line()' is deprecated as of ggplot2 3.4.0.
## i Please use the 'linewidth' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

Growth of Total NEET Population (2016 = Base Year)

Indexed NEET population changes across continents from 2016 onward



```
# Merge the NEET data with the continent mapping
merged_data <- neet_data %>%

  left_join(continents %>% select(Entity, Continent), by = "Entity")

# Filter for Asian countries and the years 2016 and 2021
asia_data <- merged_data %>%

  filter(Continent == "Asia", Year %in% c(2016, 2021), !is.na(Share))

# Identify countries with data for both 2016 and 2021
complete_asia_countries <- asia_data %>%

  group_by(Entity) %>%

  summarise(year_count = n_distinct(Year)) %>%

  filter(year_count == 2) %>%
```

```

pull(Entity)

# Filter the dataset for these countries
filtered_asia_data <- asia_data %>%
  filter(Entity %in% complete_asia_countries)

# Calculate the percentage change for each country
percentage_change <- filtered_asia_data %>%
  group_by(Entity) %>%
  summarise(
    Share_2016 = Share[Year == 2016],
    Share_2021 = Share[Year == 2021],
    Change = ((Share_2021 - Share_2016) / Share_2016) * 100
  )

# Calculate the Asia average percentage change
asia_average <- filtered_asia_data %>%
  group_by(Year) %>%
  summarise(Asia_Avg_Share = mean(Share, na.rm = TRUE)) %>%
  summarise(
    Share_2016 = Asia_Avg_Share[Year == 2016],
    Share_2021 = Asia_Avg_Share[Year == 2021],
    Change = ((Share_2021 - Share_2016) / Share_2016) * 100
  ) %>%
  pull(Change)

# Add Asia average to the dataset

```

```

percentage_change_asia <- percentage_change %>%

  add_row(Entity = "Asia Average", Change = asia_average)

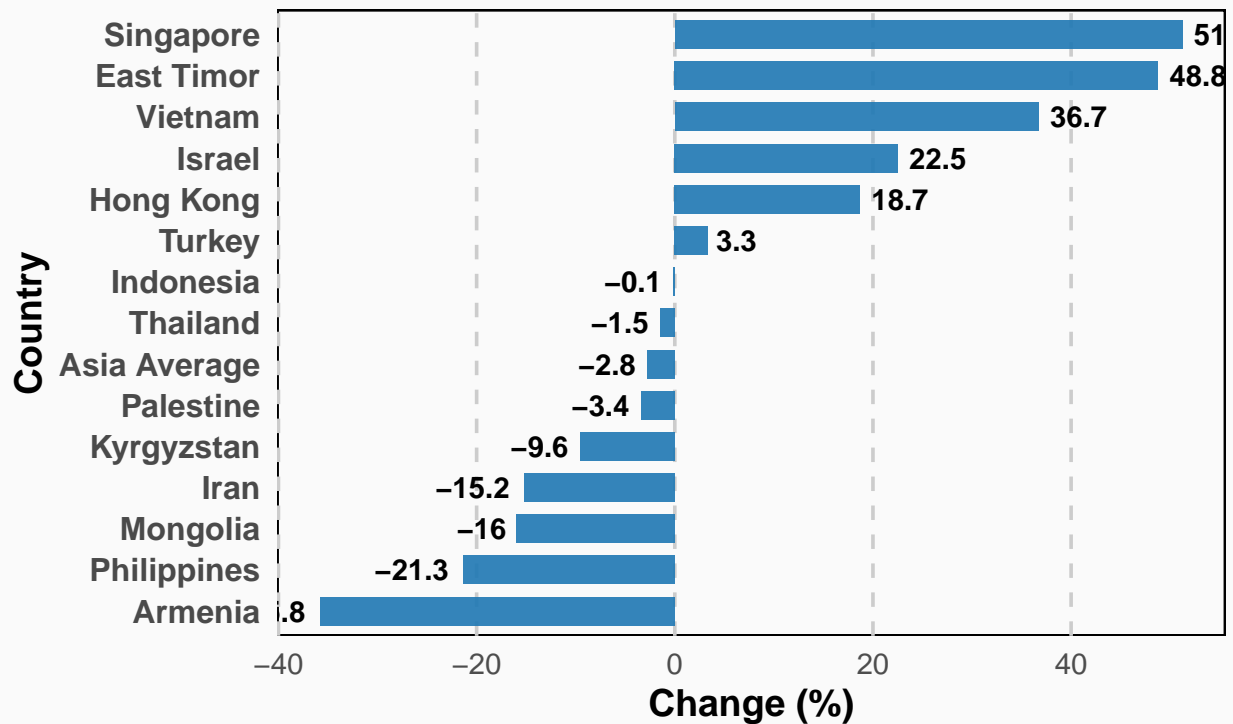
# Create the enhanced bar graph for Asia (excluding the specified country)

ggplot(percent_change_asia, aes(x = reorder(Entity, Change), y = Change)) +
  geom_bar(stat = "identity", fill = "#1F78B4", width = 0.7, alpha = 0.9) + # Vibrant blue with slight
  geom_text(aes(label = round(Change, 1)),
    hjust = ifelse(percent_change_asia$Change < 0, 1.2, -0.2),
    size = 4, color = "black", fontface = "bold") +
  coord_flip() +
  labs(
    title = "NEET Percentage Change (2016-2021) in Asia",
    subtitle = "Countries with available data for both years",
    x = "Country",
    y = "Change (%)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.y = element_text(face = "bold", color = "#4A4A4A", size = 12), # Bold country names with
    axis.title.y = element_text(face = "bold", size = 14),
    axis.title.x = element_text(face = "bold", size = 14),
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "#333333"), # Larger and
    plot.subtitle = element_text(size = 12, hjust = 0.5, face = "italic", color = "#555555"),
    panel.grid.major.y = element_blank(),
    panel.grid.major.x = element_line(color = "gray80", linetype = "dashed"), # Subtle vertical grid l
    panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "#FAFAFA", color = NA), # Light gray background
    panel.background = element_rect(fill = "#FAFAFA"),
    legend.position = "none" # Remove legend if unnecessary
  )

```

NEET Percentage Change (2016–2021) in Asia

Countries with available data for both years



```
# Filter for European countries and the years 2016 and 2021
```

```
europe_data <- merged_data %>%
```

```
  filter(Continent == "Europe", Year %in% c(2016, 2021), !is.na(Share))
```

```
# Identify countries with data for both 2016 and 2021
```

```
complete_europe_countries <- europe_data %>%
```

```
  group_by(Entity) %>%
```

```
  summarise(year_count = n_distinct(Year)) %>%
```

```
  filter(year_count == 2) %>%
```

```
  pull(Entity)
```

```
# Filter the dataset for these countries
```

```
filtered_europe_data <- europe_data %>%
```

```

filter(Entity %in% complete_europe_countries)

# Calculate the percentage change for each country
percentage_change_europe <- filtered_europe_data %>%

  group_by(Entity) %>%

  summarise(

    Share_2016 = Share[Year == 2016],

    Share_2021 = Share[Year == 2021],

    Change = ((Share_2021 - Share_2016) / Share_2016) * 100

  )

# Calculate the Europe average percentage change
europe_average <- filtered_europe_data %>%

  group_by(Year) %>%

  summarise(Europe_Avg_Share = mean(Share, na.rm = TRUE)) %>%

  summarise(

    Share_2016 = Europe_Avg_Share[Year == 2016],

    Share_2021 = Europe_Avg_Share[Year == 2021],

    Change = ((Share_2021 - Share_2016) / Share_2016) * 100

  ) %>%

  pull(Change)

# Add Europe average to the dataset
percentage_change_europe <- percentage_change_europe %>%

  add_row(Entity = "Europe Average", Change = europe_average)

# Exclude Belarus from the dataset

```

```

percentage_change_europe_filtered <- percentage_change_europe %>%

  filter(Entity != "Belarus")

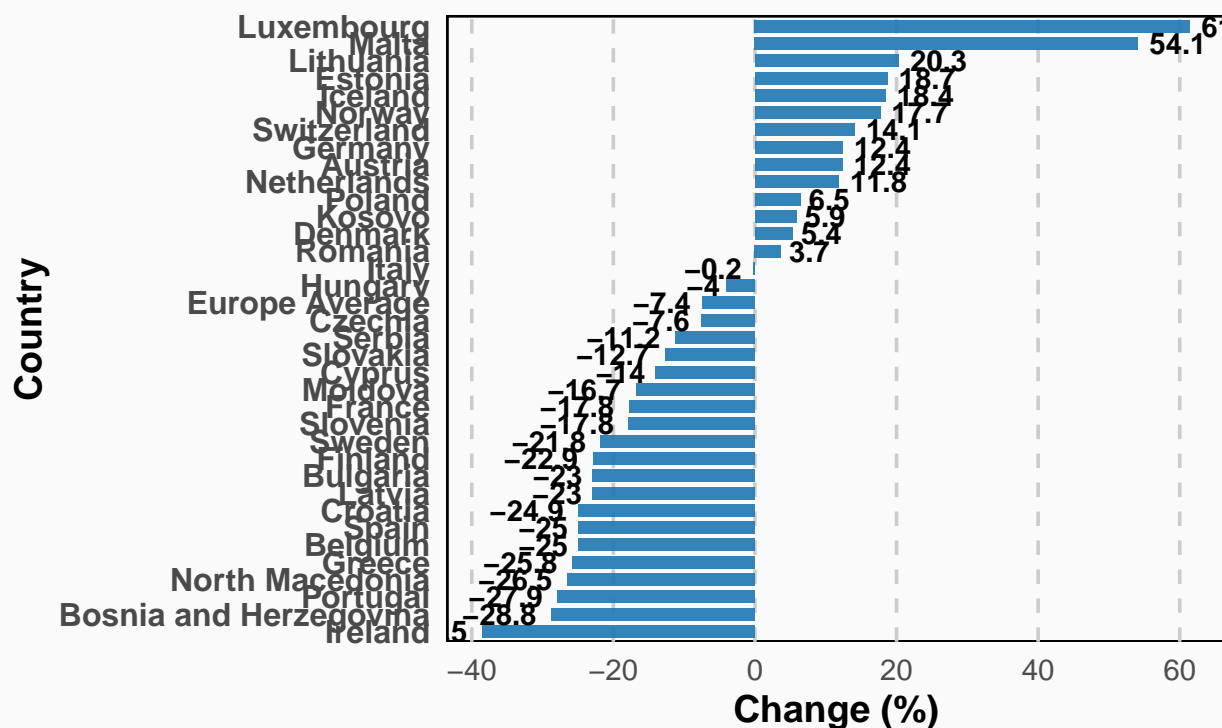
# Create the enhanced bar graph for Europe

ggplot(permission_change_europe_filtered, aes(x = reorder(Entity, Change), y = Change)) +
  geom_bar(stat = "identity", fill = "#1F78B4", width = 0.7, alpha = 0.9) + # Vibrant blue with slight
  geom_text(aes(label = round(Change, 1)),
    hjust = ifelse(permission_change_europe_filtered$Change < 0, 1.2, -0.2),
    size = 4, color = "black", fontface = "bold") +
  coord_flip() +
  labs(
    title = "NEET Percentage Change (2016-2021) in Europe",
    subtitle = "Countries with available data for both years",
    x = "Country",
    y = "Change (%)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.y = element_text(face = "bold", color = "#4A4A4A", size = 12), # Bold country names with
    axis.title.y = element_text(face = "bold", size = 14),
    axis.title.x = element_text(face = "bold", size = 14),
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "#333333"), # Larger and
    plot.subtitle = element_text(size = 12, hjust = 0.5, face = "italic", color = "#555555"),
    panel.grid.major.y = element_blank(),
    panel.grid.major.x = element_line(color = "gray80", linetype = "dashed"), # Subtle vertical grid l
    panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "#FAFAFA", color = NA), # Light gray background
    panel.background = element_rect(fill = "#FAFAFA"),
    legend.position = "none" # Remove legend if unnecessary
  )

```

NEET Percentage Change (2016–2021) in E

Countries with available data for both years



```
# Filter data for 2016 and 2021 and ensure valid Share values
```

```
filtered_data <- merged_data %>%
```

```
  filter(Year %in% c(2016, 2021), !is.na(Share))
```

```
# Identify countries with data for both 2016 and 2021
```

```
complete_countries <- filtered_data %>%
```

```
  group_by(Entity) %>%
```

```
  summarise(year_count = n_distinct(Year)) %>%
```

```
  filter(year_count == 2) %>%
```

```
  pull(Entity)
```

```
# Filter the dataset for countries with complete data
```

```
filtered_complete_data <- filtered_data %>%
```

```

filter(Entity %in% complete_countries)

# Calculate average NEET percentage for each continent in 2016 and 2021
continent_averages_2016 <- filtered_complete_data %>%
  filter(Year == 2016) %>%
  group_by(Continent) %>%
  summarise(Average_2016 = mean(Share, na.rm = TRUE))

continent_averages_2021 <- filtered_complete_data %>%
  filter(Year == 2021) %>%
  group_by(Continent) %>%
  summarise(Average_2021 = mean(Share, na.rm = TRUE))

# Merge the two datasets by continent
continent_averages <- merge(
  continent_averages_2016,
  continent_averages_2021,
  by = "Continent"
) %>%

# Remove rows with NA in any of the average columns
filter(!is.na(Average_2016) & !is.na(Average_2021)) %>%

mutate(Change = ((Average_2021 - Average_2016) / Average_2016) * 100)

# Ensure no NA in the Continent column
continent_averages <- continent_averages %>%
  filter(!is.na(Continent))

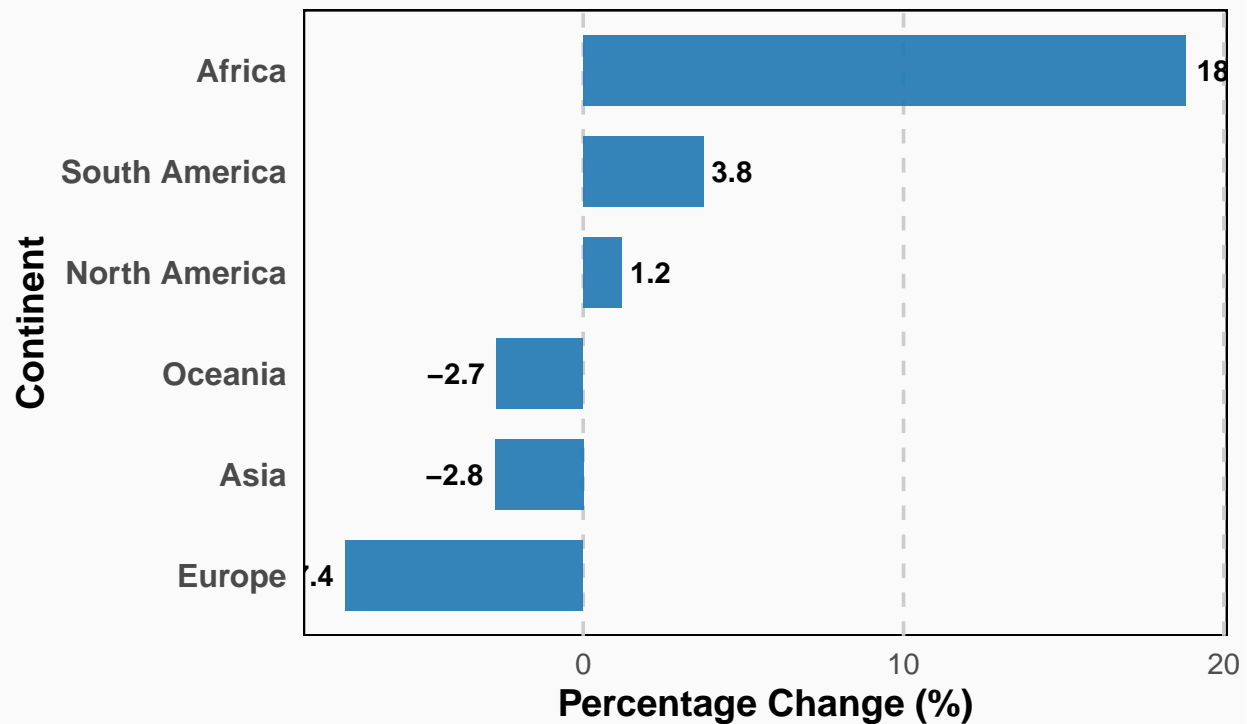
```

```
# Create the bar graph
```

```
ggplot(continent_averages, aes(x = reorder(Continent, Change), y = Change)) +  
  geom_bar(stat = "identity", fill = "#1F78B4", width = 0.7, alpha = 0.9) + # Vibrant blue with slight  
  geom_text(aes(label = round(Change, 1)),  
            hjust = ifelse(continent_averages$Change < 0, 1.2, -0.2),  
            size = 4, color = "black", fontface = "bold") + # Bold and clear labels  
  coord_flip() + # Horizontal bar chart  
  labs(  
    title = "NEET Percentage Change (2016-2021) by Continent",  
    subtitle = "Averages calculated for countries with data available for both years",  
    x = "Continent",  
    y = "Percentage Change (%)"  
  ) +  
  theme_minimal(base_size = 14) +  
  theme(  
    axis.text.y = element_text(face = "bold", color = "#4A4A4A", size = 12), # Bold continent names wi  
    axis.title.y = element_text(face = "bold", size = 14),  
    axis.title.x = element_text(face = "bold", size = 14),  
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "#333333"), # Larger and  
    plot.subtitle = element_text(size = 12, hjust = 0.5, face = "italic", color = "#555555"), # Subtle  
    panel.grid.major.y = element_blank(), # Remove horizontal grid lines  
    panel.grid.major.x = element_line(color = "gray80", linetype = "dashed"), # Dashed vertical grid l  
    panel.grid.minor = element_blank(), # Remove minor grid lines  
    plot.background = element_rect(fill = "#FAFAFA", color = NA), # Light gray background  
    panel.background = element_rect(fill = "#FAFAFA"),  
    legend.position = "none" # Remove legend if unnecessary  
  )
```

NEET Percentage Change (2016–2021) by Continent

Averages calculated for countries with data available for both years



```
# Filter data for 2016 and 2021, ensuring valid Share and Continent values
```

```
filtered_data <- merged_data %>%
```

```
  filter(Year %in% c(2016, 2021), !is.na(Share), !is.na(Continent)) # Ensure no NA values for Share or Continent
```

```
# Identify countries with data for both 2016 and 2021
```

```
complete_countries <- filtered_data %>%
```

```
  group_by(Entity) %>%
```

```
  summarise(year_count = n_distinct(Year)) %>%
```

```
  filter(year_count == 2) %>%
```

```
  pull(Entity)
```

```
# Filter the dataset for countries with data for both years
```

```
filtered_complete_data <- filtered_data %>%
```

```

filter(Entity %in% complete_countries)

# Create a combined boxplot for both 2016 and 2021 using facet_wrap

ggplot(filtered_complete_data, aes(x = Continent, y = Share, fill = Continent)) +
  geom_boxplot(outlier.color = "red", outlier.shape = 16, outlier.size = 2, alpha = 0.8) + # Highlight ti
  facet_wrap(~ Year, scales = "free_y", ncol = 2) + # Two columns for better space utilization
  scale_fill_brewer(palette = "Pastell1") + # Softer color palette for better readability
  labs(
    title = "NEET Percentage Distribution by Continent (2016 vs 2021)",
    subtitle = "Faceted by Year for a comparative view",
    x = "Continent",
    y = "NEET Percentage (%)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, color = "#4A4A4A", size = 10), # Rotate and styl
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "#333333"), # Enhanced ti
    plot.subtitle = element_text(face = "italic", size = 12, hjust = 0.5, color = "#555555"), # Subtle
    axis.title = element_text(face = "bold", size = 12, color = "#333333"),
    panel.grid.major.y = element_line(color = "gray80", size = 0.5, linetype = "dashed"), # Dashed y-a
    panel.grid.major.x = element_blank(), # No vertical grid lines
    panel.grid.minor = element_blank(),
    strip.text = element_text(face = "bold", size = 12, color = "white"), # Bold facet labels
    strip.background = element_rect(fill = "#1F78B4", color = NA), # Dark blue background for facet la
    legend.position = "none", # Remove legend since it's redundant
    plot.background = element_rect(fill = "#F9F9F9", color = NA), # Light background for a clean look
    panel.background = element_rect(fill = "#F9F9F9", color = NA)
  )

```

NEET Percentage Distribution by Continent (2016 vs 2021)

Faceted by Year for a comparative view

