analysis.R

abdulbasir

2025-05-13

```
######################################################################
# Global Poverty Analysis
# STA 3000 Final Project
######################################################################

######################################################################
# Research Hypotheses and Equations
######################################################################

# Primary Research Hypotheses:
# H1: Countries with lower income inequality (measured by mean-to-median ratio)
#     will show greater poverty reduction over time
# H2: The relationship between economic growth and poverty reduction is moderated
#     by income distribution patterns
# H3: Regional differences in poverty reduction success are statistically significant
# H4: The poorest decile's growth rate is a significant predictor of overall poverty reduction

# Key Equations:

# 1. Inequality Metrics
# Mean-to-Median Ratio = Mean_Income / Median_Income
# Richest-to-Poorest Ratio = Richest_Decile / Poorest_Decile
# Gini Coefficient = 1 - (2/n) * sum((n-i+0.5)/n * y_i)

# 2. Poverty Reduction Metrics
# Absolute Reduction = Initial_Poverty - Final_Poverty
# Relative Reduction = (Initial_Poverty - Final_Poverty) / Initial_Poverty * 100
# Annual Reduction Rate = Absolute_Reduction / Year_Range

# 3. Statistical Models
# Model 1: Linear relationship between income and poverty
# Extreme_Poverty_Share = b0 + b1*log(Mean_Income) + e

# Model 2: Inequality's effect on poverty
# Extreme_Poverty_Share = b0 + b1*Richest_to_Poorest_Ratio + e

# Model 3: Combined effects
# Extreme_Poverty_Share = b0 + b1*log(Mean_Income) + b2*Richest_to_Poorest_Ratio + e

# Model 4: Interaction effects
# Extreme_Poverty_Share = b0 + b1*log(Mean_Income) + b2*Richest_to_Poorest_Ratio +
#                         b3*(log(Mean_Income)*Richest_to_Poorest_Ratio) + e
```

```r
# 4. Time Series Components
# Y(t) = T(t) + S(t) + R(t)
# where:
# Y(t) = Observed poverty rate at time t
# T(t) = Trend component
# S(t) = Seasonal component (if any)
# R(t) = Random component

# 5. Growth Incidence Analysis
# Growth_Rate = (Final_Value - Initial_Value) / Initial_Value * 100
# Decile_Growth_Ratio = Growth_Rate_Poorest_Decile / Growth_Rate_Richest_Decile

# loading necessary packages
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
```

```r
library(ggplot2)
library(tidyr)
library(scales)
library(lmtest)      # For dwtest
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(car)         # For vif
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(forecast)    # For auto.arima and forecast
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(tseries)     # For time series functions
library(tibble)      # For rownames_to_column

######################################################################
# Part 1: Data Loading And Exploration
######################################################################

# setting directory
getwd()
```

```
## [1] "/Users/abdulbasir/income-distribution-poverty-dynamics/scripts"
```

```r
# use relative paths instead of setwd
data_dir = "../data"
output_dir = "../output"
viz_dir = "../output/visualizations"

# reading the six datasets
mean_income = read.csv(file.path(data_dir, "mean-income-or-consumption-per-day.csv"))
median_income = read.csv(file.path(data_dir, "median-income-or-consumption-per-day.csv"))
poorest_decile = read.csv(file.path(data_dir, "the-poorest-decile.csv"))
richest_decile = read.csv(file.path(data_dir, "the-richest-decile.csv"))
num_below_poverty = read.csv(file.path(data_dir, "number-of-people-living-below-a-range-of-poverty-lines
share_below_poverty = read.csv(file.path(data_dir, "share-of-population-living-below-a-range-of-poverty-

# basic exploration of each dataset
head(mean_income)
```

```
##   Country Year Mean.income.or.consumption.per.day
## 1 Albania 1996                           7.933157
## 2 Albania 2002                           8.108229
## 3 Albania 2005                           9.165975
## 4 Albania 2008                          10.038169
## 5 Albania 2012                           9.517231
## 6 Albania 2014                          10.141310
```

```r
summary(mean_income)
```

```
##    Country               Year      Mean.income.or.consumption.per.day
##  Length:2705        Min.   :1963   Min.   : 0.997
##  Class :character   1st Qu.:1999   1st Qu.: 6.901
##  Mode  :character   Median :2008   Median :14.157
##                     Mean   :2006   Mean   :22.586
##                     3rd Qu.:2015   3rd Qu.:35.254
##                     Max.   :2024   Max.   :93.328
##                                    NA's   :70
```

```r
dim(mean_income)
```

```
## [1] 2705    3
```

```r
head(median_income)
```

```
##   Country Year Median.income.or.consumption.per.day
## 1 Albania 1996                             6.972103
## 2 Albania 2002                             6.688141
## 3 Albania 2005                             7.799791
## 4 Albania 2008                             8.400200
## 5 Albania 2012                             8.240385
## 6 Albania 2014                             8.295376
```

```r
summary(median_income)
```

```
##    Country               Year      Median.income.or.consumption.per.day
##  Length:2705        Min.   :1963   Min.   : 0.690
##  Class :character   1st Qu.:1999   1st Qu.: 4.836
##  Mode  :character   Median :2008   Median :10.274
##                     Mean   :2006   Mean   :18.144
##                     3rd Qu.:2015   3rd Qu.:29.487
##                     Max.   :2024   Max.   :79.700
##                                    NA's   :70
```

```r
dim(median_income)
```

```
## [1] 2705    3
```

```r
head(poorest_decile)
```

```
##   Country Year
## 1 Albania 1996
## 2 Albania 2002
## 3 Albania 2005
## 4 Albania 2008
## 5 Albania 2012
## 6 Albania 2014
##   Threshold.income.or.consumption.per.day.marking.the.poorest.decile
## 1                                                           3.691232
## 2                                                           3.501889
## 3                                                           3.983527
## 4                                                           4.598464
## 5                                                           4.410012
## 6                                                           3.672949
```

```r
summary(poorest_decile)
```

```
##    Country               Year
##  Length:2705        Min.   :1963
##  Class :character   1st Qu.:1999
##  Mode  :character   Median :2008
##                     Mean   :2006
##                     3rd Qu.:2015
##                     Max.   :2024
##
##  Threshold.income.or.consumption.per.day.marking.the.poorest.decile
##  Min.   : 0.250
##  1st Qu.: 1.858
##  Median : 4.035
##  Mean   : 8.374
##  3rd Qu.:13.483
##  Max.   :37.034
##  NA's   :72
```

```r
dim(poorest_decile)
```

```
## [1] 2705    3
```

```r
head(richest_decile)
```

```
##   Country Year
## 1 Albania 1996
## 2 Albania 2002
## 3 Albania 2005
## 4 Albania 2008
## 5 Albania 2012
## 6 Albania 2014
##   Threshold.income.or.consumption.per.day.marking.the.richest.decile
## 1                                                           13.16480
## 2                                                           13.99076
## 3                                                           15.43352
## 4                                                           16.43590
## 5                                                           16.00902
## 6                                                           19.05701
```

```r
summary(richest_decile)
```

```
##    Country               Year
##  Length:2705        Min.   :1963
##  Class :character   1st Qu.:1999
##  Mode  :character   Median :2008
##                     Mean   :2006
##                     3rd Qu.:2015
##                     Max.   :2024
##
##  Threshold.income.or.consumption.per.day.marking.the.richest.decile
##  Min.   :  1.578
##  1st Qu.: 12.457
##  Median : 27.802
```

```
##  Mean   : 40.131
##  3rd Qu.: 60.972
##  Max.   :167.603
##  NA's   :72
```

```r
dim(richest_decile)
```

```
## [1] 2705    3
```

```r
head(num_below_poverty)
```

```
##   Country Year Number.below..1.a.day Number.below..2.15.a.day
## 1 Albania 1996                  1819                    16944
## 2 Albania 2002                  1415                    33337
## 3 Albania 2005                     0                    17800
## 4 Albania 2008                     0                     5892
## 5 Albania 2012                   699                    18003
## 6 Albania 2014                     0                    29558
##   Number.below..3.65.a.day Number.below..6.85.a.day Number.below..10.a.day
## 1                   281906                  1516827                2395790
## 2                   347459                  1589301                2340278
## 3                   219516                  1217497                2061456
## 4                   115022                   985342                1849996
## 5                   139377                  1063512                1887070
## 6                   279549                  1119350                1761722
##   Number.below..20.a.day Number.below..30.a.day Number.below..40.a.day
## 1                3126915                3164784                3166502
## 2                2945105                3018917                3043749
## 3                2873129                2977076                2997843
## 4                2787653                2903376                2927993
## 5                2770468                2873530                2892932
## 6                2645988                2834178                2874814
```

```r
summary(num_below_poverty)
```

```
##    Country               Year      Number.below..1.a.day
##  Length:2705        Min.   :1963   Min.   :        0
##  Class :character   1st Qu.:1999   1st Qu.:     1982
##  Mode  :character   Median :2008   Median :    73331
##                     Mean   :2006   Mean   : 14547048
##                     3rd Qu.:2015   3rd Qu.:  1290650
##                     Max.   :2024   Max.   :480961600
##  Number.below..2.15.a.day Number.below..3.65.a.day Number.below..6.85.a.day
##  Min.   :0.000e+00        Min.   :0.000e+00        Min.   :0.000e+00
##  1st Qu.:2.500e+04        1st Qu.:1.060e+05        1st Qu.:3.759e+05
##  Median :4.014e+05        Median :1.208e+06        Median :3.521e+06
##  Mean   :7.474e+07        Mean   :1.457e+08        Mean   :2.217e+08
##  3rd Qu.:6.835e+06        3rd Qu.:1.572e+07        3rd Qu.:3.544e+07
##  Max.   :2.011e+09        Max.   :3.177e+09        Max.   :4.275e+09
##  Number.below..10.a.day Number.below..20.a.day Number.below..30.a.day
##  Min.   :0.000e+00      Min.   :2.875e+03      Min.   :1.034e+04
##  1st Qu.:7.763e+05      1st Qu.:2.472e+06      1st Qu.:3.412e+06
```

```
##  Median :5.173e+06      Median :9.173e+06      Median :1.390e+07
##  Mean   :2.583e+08      Mean   :3.088e+08      Mean   :3.316e+08
##  3rd Qu.:4.541e+07      3rd Qu.:8.299e+07      3rd Qu.:9.219e+07
##  Max.   :4.764e+09      Max.   :6.089e+09      Max.   :6.702e+09
##  Number.below..40.a.day
##  Min.   :1.043e+04
##  1st Qu.:3.909e+06
##  Median :1.604e+07
##  Mean   :3.472e+08
##  3rd Qu.:1.071e+08
##  Max.   :7.065e+09
```

```r
dim(num_below_poverty)
```

```
## [1] 2705   10
```

```r
head(share_below_poverty)
```

```
##            Country Year Share.below..1.a.day Share.below..2.15.a.day
## 1          Albania 2016            1.7444307                5.795102
## 2          Albania 2017            1.1722292                5.264806
## 3          Albania 2018            0.7443276                3.892983
## 4 Argentina (urban) 1980            0.0000000                0.000000
## 5 Argentina (urban) 1986            1.1022475                1.119132
## 6 Argentina (urban) 1987            0.7880118                1.186086
##   Share.below..3.65.a.day Share.below..6.85.a.day Share.below..10.a.day
## 1               16.190832               41.567802              62.19595
## 2               13.959268               37.323772              59.01644
## 3               11.321431               34.187016              55.64245
## 4                0.000000                5.701000              13.29800
## 5                1.827732                4.770096              10.25071
## 6                2.093658                7.622199              15.59101
##   Share.below..20.a.day Share.below..30.a.day Share.below..40.a.day
## 1              90.78622              97.44051              99.13977
## 2              90.26276              97.62189              99.29549
## 3              89.31027              97.11217              99.20074
## 4              38.75100              60.28000              74.65900
## 5              36.40442              58.63238              72.93964
## 6              43.53422              63.70861              76.84965
```

```r
summary(share_below_poverty)
```

```
##    Country               Year      Share.below..1.a.day Share.below..2.15.a.day
##  Length:1468        Min.   :1963   Min.   : 0.00000     Min.   : 0.0000
##  Class :character   1st Qu.:1998   1st Qu.: 0.03359     1st Qu.: 0.1562
##  Mode  :character   Median :2008   Median : 0.24916     Median : 0.6158
##                     Mean   :2006   Mean   : 1.33230     Mean   : 3.9166
##                     3rd Qu.:2015   3rd Qu.: 0.86040     3rd Qu.: 3.5207
##                     Max.   :2023   Max.   :59.25600     Max.   :96.8710
##  Share.below..3.65.a.day Share.below..6.85.a.day Share.below..10.a.day
##  Min.   :  0.000         Min.   :  0.0000        Min.   :  0.000
##  1st Qu.:  0.249         1st Qu.:  0.7459        1st Qu.:  1.482
```

```
##  Median :  1.217         Median :  3.5407       Median :  8.584  
##  Mean   :  7.845         Mean   : 16.3070       Mean   : 23.828  
##  3rd Qu.:  9.940         3rd Qu.: 28.3458       3rd Qu.: 46.435  
##  Max.   :100.000         Max.   :100.0000       Max.   :100.000  
##  Share.below..20.a.day Share.below..30.a.day Share.below..40.a.day
##  Min.   :  0.7234      Min.   :  5.955       Min.   : 12.86       
##  1st Qu.:  9.2491      1st Qu.: 26.748       1st Qu.: 45.95       
##  Median : 39.9696      Median : 66.470       Median : 81.05       
##  Mean   : 43.3496      Mean   : 58.593       Mean   : 70.47       
##  3rd Qu.: 78.7209      3rd Qu.: 89.937       3rd Qu.: 94.57       
##  Max.   :100.0000      Max.   :100.000       Max.   :100.00
```

```r
dim(share_below_poverty)
```

```
## [1] 1468   10
```

```r
# checking for missing values
sum(is.na(mean_income))
```

```
## [1] 70
```

```r
sum(is.na(median_income))
```

```
## [1] 70
```

```r
sum(is.na(poorest_decile))
```

```
## [1] 72
```

```r
sum(is.na(richest_decile))
```

```
## [1] 72
```

```r
sum(is.na(num_below_poverty))
```

```
## [1] 0
```

```r
sum(is.na(share_below_poverty))
```

```
## [1] 0
```

```r
#####################################################################
# Part 2: Data Integration
#####################################################################

# merging income distribution metrics (mean, median, poorest and richest deciles)
income_distribution = mean_income %>%
  inner_join(median_income, by = c("Country", "Year")) %>%
```

```r
  inner_join(poorest_decile, by = c("Country", "Year")) %>%
  inner_join(richest_decile, by = c("Country", "Year"))

# renaming columns for clarity
names(income_distribution) = c("Country", "Year", "Mean_Income", "Median_Income",
                               "Poorest_Decile", "Richest_Decile")

# checking the merged dataset
head(income_distribution)
```

```
##   Country Year Mean_Income Median_Income Poorest_Decile Richest_Decile
## 1 Albania 1996    7.933157      6.972103       3.691232       13.16480
## 2 Albania 2002    8.108229      6.688141       3.501889       13.99076
## 3 Albania 2005    9.165975      7.799791       3.983527       15.43352
## 4 Albania 2008   10.038169      8.400200       4.598464       16.43590
## 5 Albania 2012    9.517231      8.240385       4.410012       16.00902
## 6 Albania 2014   10.141310      8.295376       3.672949       19.05701
```

```r
dim(income_distribution)
```

```
## [1] 2705    6
```

```r
# calculating inequality metrics
income_distribution = income_distribution %>%
  mutate(
    Mean_to_Median_Ratio = Mean_Income / Median_Income,
    Richest_to_Poorest_Ratio = Richest_Decile / Poorest_Decile,
    Top_Bottom_Gap = Richest_Decile - Poorest_Decile
  )

# checking our inequality metrics
summary(income_distribution$Mean_to_Median_Ratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.046   1.184   1.270   1.359   1.431   3.929      70
```

```r
summary(income_distribution$Richest_to_Poorest_Ratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   2.286   3.989   5.251   6.644   7.524 115.658      72
```

```r
summary(income_distribution$Top_Bottom_Gap)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.081   9.934  23.520  31.757  47.750 139.740      72
```

```r
###################################################################
# Part 3: Correlation Analysis
###################################################################
```

```r
# step 1: prepare correlation data
# first merge poverty data with income distribution data
poverty_income = income_distribution %>%
  inner_join(share_below_poverty %>%
              dplyr::select(Country, Year, "Share.below..2.15.a.day") %>%
              rename(Extreme_Poverty_Share = "Share.below..2.15.a.day"),
            by = c("Country", "Year"))

# now prepare correlation data
correlation_data = poverty_income %>%
  dplyr::select(
    Extreme_Poverty_Share,
    Mean_Income,
    Richest_to_Poorest_Ratio,
    Mean_to_Median_Ratio
  )

# step 2: calculate correlation matrix
correlation_matrix = cor(correlation_data, use = "complete.obs")
print("correlation matrix:")
```

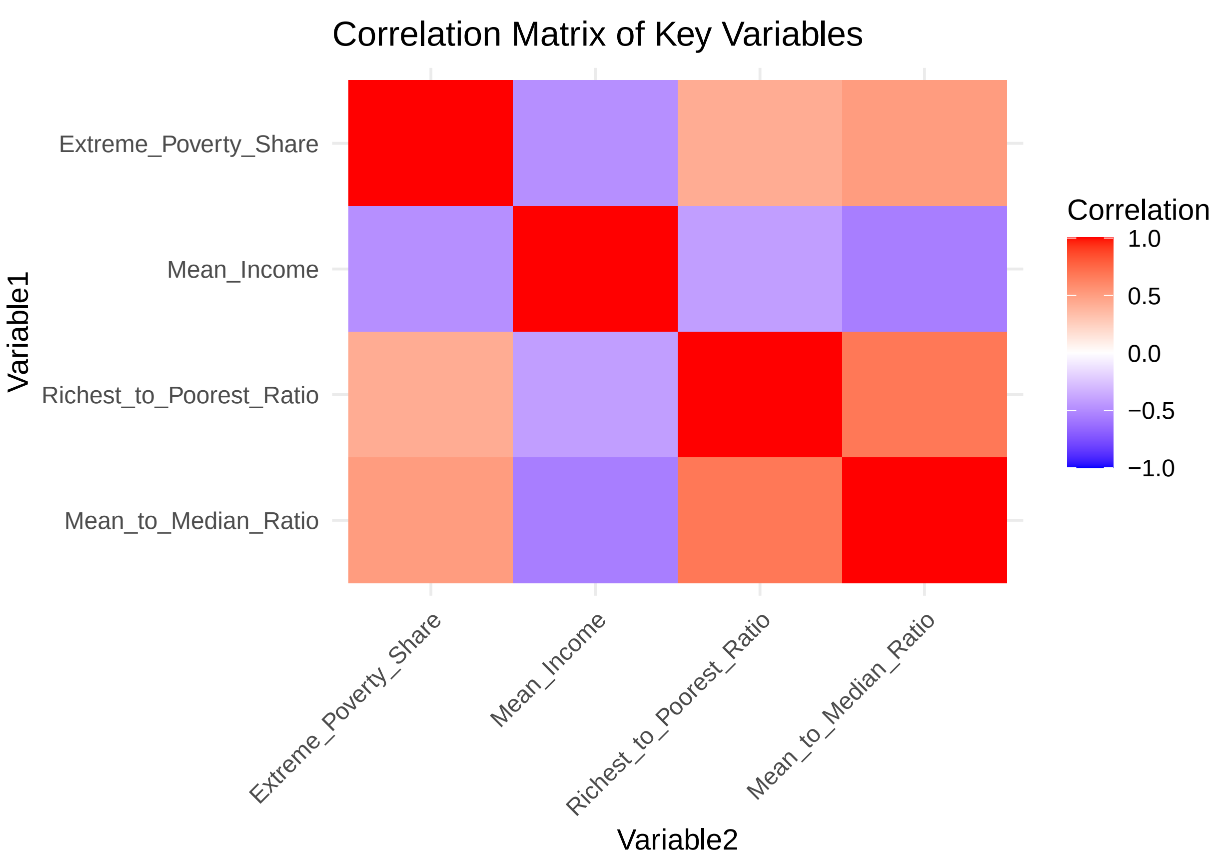
```
## [1] "correlation matrix:"
```

```r
print(correlation_matrix)
```

```
##                          Extreme_Poverty_Share Mean_Income
## Extreme_Poverty_Share                1.0000000  -0.4812393
## Mean_Income                         -0.4812393   1.0000000
## Richest_to_Poorest_Ratio             0.4278347  -0.4174949
## Mean_to_Median_Ratio                 0.5079167  -0.5588057
##                          Richest_to_Poorest_Ratio Mean_to_Median_Ratio
## Extreme_Poverty_Share                   0.4278347            0.5079167
## Mean_Income                            -0.4174949           -0.5588057
## Richest_to_Poorest_Ratio                1.0000000            0.6771407
## Mean_to_Median_Ratio                    0.6771407            1.0000000
```

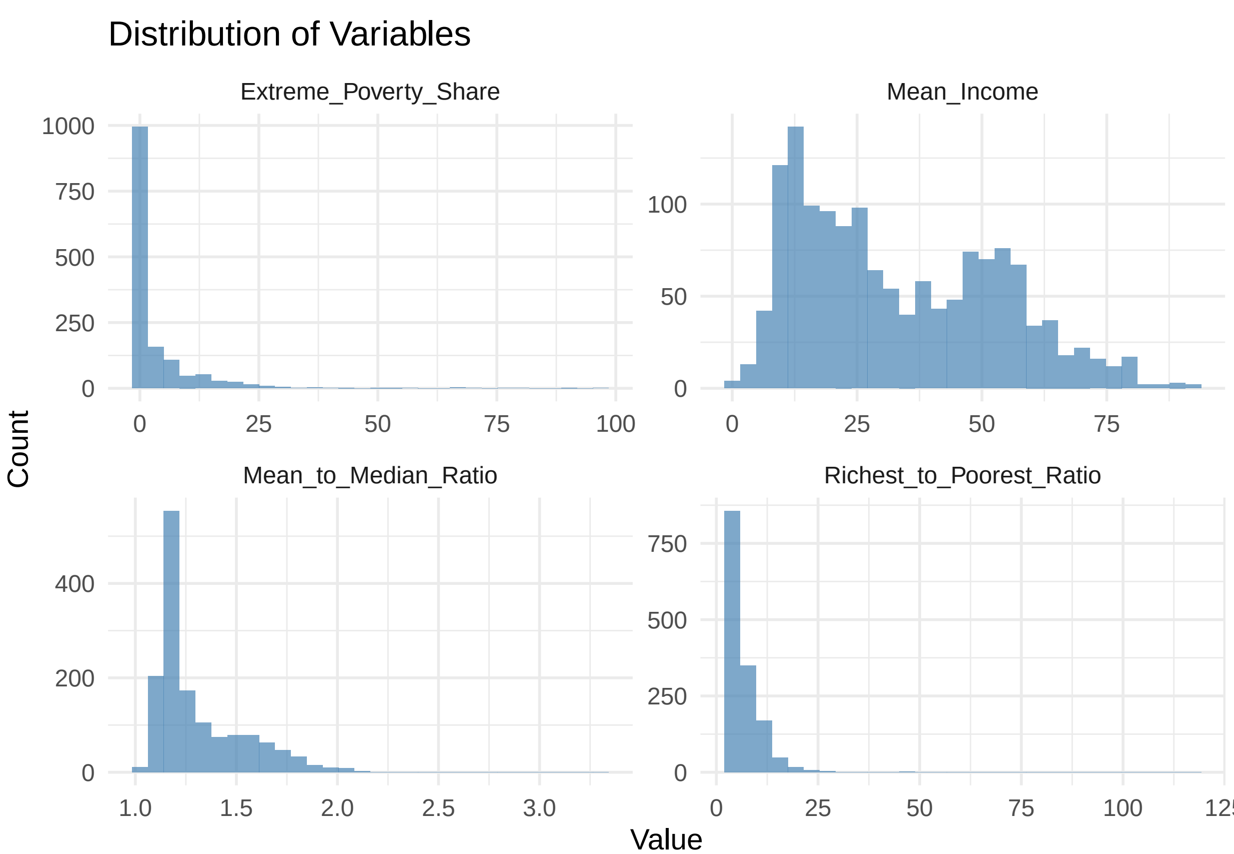
```r
# step 3: correlation significance tests
correlation_tests = list()
for(i in 1:ncol(correlation_data)) {
  for(j in 1:ncol(correlation_data)) {
    if(i != j) {
      test = cor.test(correlation_data[,i], correlation_data[,j])
      correlation_tests[[paste(colnames(correlation_data)[i],
                               colnames(correlation_data)[j],
                               sep = "_")]] = test
    }
  }
}

# step 4: visualize correlations
# prepare correlation data for ggplot
correlation_long = as.data.frame(correlation_matrix) %>%
```

```r
  rownames_to_column("Variable1") %>%
  pivot_longer(-Variable1, names_to = "Variable2", values_to = "Correlation") %>%
  mutate(
    Variable1 = factor(Variable1, levels = rev(rownames(correlation_matrix))),
    Variable2 = factor(Variable2, levels = colnames(correlation_matrix))
  )

# create correlation heatmap using ggplot2
correlation_heatmap = ggplot(correlation_long, aes(x = Variable2, y = Variable1, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Matrix of Key Variables")

# save correlation heatmap
ggsave(file.path(viz_dir, "correlation_heatmap.png"),
       correlation_heatmap,
       width = 10,
       height = 8,
       dpi = 300)
print(correlation_heatmap)
```




```r
# create scatter plot matrix using ggplot2
# prepare data for scatter plot matrix
```

```r
scatter_data = correlation_data %>%
  pivot_longer(everything(),
               names_to = "Variable",
               values_to = "Value")

# create scatter plot matrix
scatter_matrix = ggplot(scatter_data, aes(x = Value)) +
  geom_histogram(bins = 30, fill = "steelblue", alpha = 0.7) +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribution of Variables",
       x = "Value",
       y = "Count")

# save scatter matrix
ggsave(file.path(viz_dir, "variable_distributions.png"),
       scatter_matrix,
       width = 12,
       height = 8,
       dpi = 300)

# print the scatter matrix
print(scatter_matrix)
```




```r
# step 5: partial correlations
# install and load ppcor if not already installed
```

```r
if (!require("ppcor")) {
  install.packages("ppcor")
  library(ppcor)
}
```

```
## Loading required package: ppcor

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
partial_cor = pcor(correlation_data)
print("Partial Correlation Matrix:")
```

```
## [1] "Partial Correlation Matrix:"
```

```r
print(partial_cor$estimate)
```

```
##                         Extreme_Poverty_Share Mean_Income
## Extreme_Poverty_Share               1.0000000 -0.27080002
## Mean_Income                        -0.2708000  1.00000000
## Richest_to_Poorest_Ratio            0.1195507 -0.02886963
## Mean_to_Median_Ratio                0.2077998 -0.33254544
##                         Richest_to_Poorest_Ratio Mean_to_Median_Ratio
## Extreme_Poverty_Share                 0.11955068            0.2077998
## Mean_Income                          -0.02886963           -0.3325454
## Richest_to_Poorest_Ratio              1.00000000            0.5471574
## Mean_to_Median_Ratio                  0.54715738            1.0000000
```

```r
# step 6: non-parametric correlations
spearman_cor = cor(correlation_data, method = "spearman")
print("spearman correlation matrix:")
```

```
## [1] "spearman correlation matrix:"
```

```r
print(spearman_cor)
```

```
##                         Extreme_Poverty_Share Mean_Income
## Extreme_Poverty_Share               1.0000000  -0.7436502
## Mean_Income                        -0.7436502   1.0000000
## Richest_to_Poorest_Ratio            0.8051110  -0.6314107
## Mean_to_Median_Ratio                0.6652683  -0.5551897
##                         Richest_to_Poorest_Ratio Mean_to_Median_Ratio
## Extreme_Poverty_Share                  0.8051110            0.6652683
## Mean_Income                           -0.6314107           -0.5551897
## Richest_to_Poorest_Ratio               1.0000000            0.8982787
## Mean_to_Median_Ratio                   0.8982787            1.0000000
```

```r
# step 7: correlation by region
regional_correlations = list()
for(region in unique(poverty_income$Region)) {
  region_data = poverty_income %>%
    filter(Region == region) %>%
    dplyr::select(
      Extreme_Poverty_Share,
      Mean_Income,
      Richest_to_Poorest_Ratio,
      Mean_to_Median_Ratio
    )

  if(nrow(region_data) > 1) {
    regional_correlations[[region]] = cor(region_data, use = "complete.obs")
  }
}

# step 8: correlation stability analysis
# calculate rolling correlations
library(zoo)
rolling_cor = rollapply(correlation_data,
                        width = 10,
                        function(x) cor(x)[1,2],
                        by.column = FALSE)
```

```
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
## Warning in cor(x): the standard deviation is zero
```

```r
# step 9: results summary
cat("\ncorrelation analysis results:\n")
```

```
##
## correlation analysis results:
```

```r
cat("1. pearson correlations:\n")
```

```
## 1. pearson correlations:
```

```r
for(i in 1:ncol(correlation_matrix)) {
  for(j in 1:ncol(correlation_matrix)) {
    if(i < j) {
      cat("   -", colnames(correlation_matrix)[i], "vs",
          colnames(correlation_matrix)[j], ":",
          round(correlation_matrix[i,j], 3), "\n")
    }
  }
}
```

```
##    - Extreme_Poverty_Share vs Mean_Income : -0.481
##    - Extreme_Poverty_Share vs Richest_to_Poorest_Ratio : 0.428
##    - Extreme_Poverty_Share vs Mean_to_Median_Ratio : 0.508
##    - Mean_Income vs Richest_to_Poorest_Ratio : -0.417
##    - Mean_Income vs Mean_to_Median_Ratio : -0.559
##    - Richest_to_Poorest_Ratio vs Mean_to_Median_Ratio : 0.677
```

```r
cat("\n2. significant correlations (p < 0.05):\n")
```

```
##
## 2. significant correlations (p < 0.05):
```

```r
for(test in names(correlation_tests)) {
  if(correlation_tests[[test]]$p.value < 0.05) {
    cat("   -", test, ":",
        round(correlation_tests[[test]]$estimate, 3),
        "(p =", round(correlation_tests[[test]]$p.value, 3), ")\n")
  }
}
```

```
##    - Extreme_Poverty_Share_Mean_Income : -0.481 (p = 0 )
##    - Extreme_Poverty_Share_Richest_to_Poorest_Ratio : 0.428 (p = 0 )
##    - Extreme_Poverty_Share_Mean_to_Median_Ratio : 0.508 (p = 0 )
##    - Mean_Income_Extreme_Poverty_Share : -0.481 (p = 0 )
##    - Mean_Income_Richest_to_Poorest_Ratio : -0.417 (p = 0 )
##    - Mean_Income_Mean_to_Median_Ratio : -0.559 (p = 0 )
##    - Richest_to_Poorest_Ratio_Extreme_Poverty_Share : 0.428 (p = 0 )
##    - Richest_to_Poorest_Ratio_Mean_Income : -0.417 (p = 0 )
##    - Richest_to_Poorest_Ratio_Mean_to_Median_Ratio : 0.677 (p = 0 )
##    - Mean_to_Median_Ratio_Extreme_Poverty_Share : 0.508 (p = 0 )
##    - Mean_to_Median_Ratio_Mean_Income : -0.559 (p = 0 )
##    - Mean_to_Median_Ratio_Richest_to_Poorest_Ratio : 0.677 (p = 0 )
```

```r
cat("\n3. regional correlation patterns:\n")
```

```
##
## 3. regional correlation patterns:
```

```r
for(region in names(regional_correlations)) {
  cat("   -", region, ":\n")
  cat("     poverty-income correlation:",
      round(regional_correlations[[region]][1,2], 3), "\n")
  cat("     poverty-inequality correlation:",
      round(regional_correlations[[region]][1,3], 3), "\n")
}

cat("\n4. correlation stability:\n")
```

```
##
## 4. correlation stability:
```

```r
cat("   - rolling correlation range:",
    round(min(rolling_cor, na.rm = TRUE), 3), "to",
    round(max(rolling_cor, na.rm = TRUE), 3), "\n")
```

```
##    - rolling correlation range: -0.996 to 0.893
```

```r
cat("   - correlation volatility:",
    round(sd(rolling_cor, na.rm = TRUE), 3), "\n")
```

```
##    - correlation volatility: 0.49
```

```r
######################################################################
# Part 4: Trend Analysis Of Income Distribution
######################################################################

# function to calculate global average for each year
calculate_yearly_avg = function(data, metric) {
  yearly_avg = data %>%
    group_by(Year) %>%
    summarize(
      Average = mean(!!sym(metric), na.rm = TRUE),
      Median = median(!!sym(metric), na.rm = TRUE),
      Count = n()
    )
  return(yearly_avg)
}

# calculating global trends for inequality metrics
mean_median_trend = calculate_yearly_avg(income_distribution, "Mean_to_Median_Ratio")
rich_poor_trend = calculate_yearly_avg(income_distribution, "Richest_to_Poorest_Ratio")

# let's look at the trends
head(mean_median_trend)
```

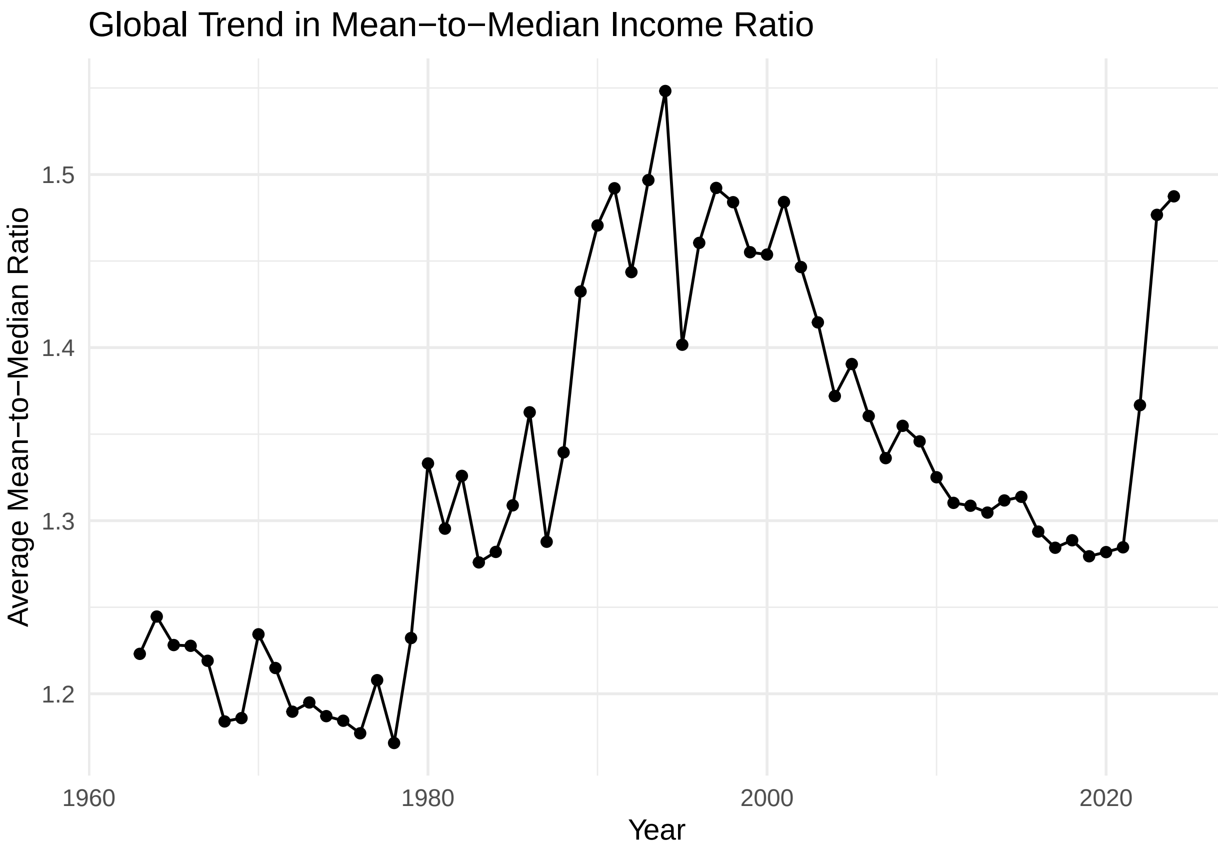
```
## # A tibble: 6 x 4
##    Year Average Median Count
##   <int>   <dbl>  <dbl> <int>
## 1  1963    1.22   1.22     1
```

```
## 2  1964    1.24   1.24     1
## 3  1965    1.23   1.23     1
## 4  1966    1.23   1.23     1
## 5  1967    1.22   1.22     1
## 6  1968    1.18   1.18     2
```

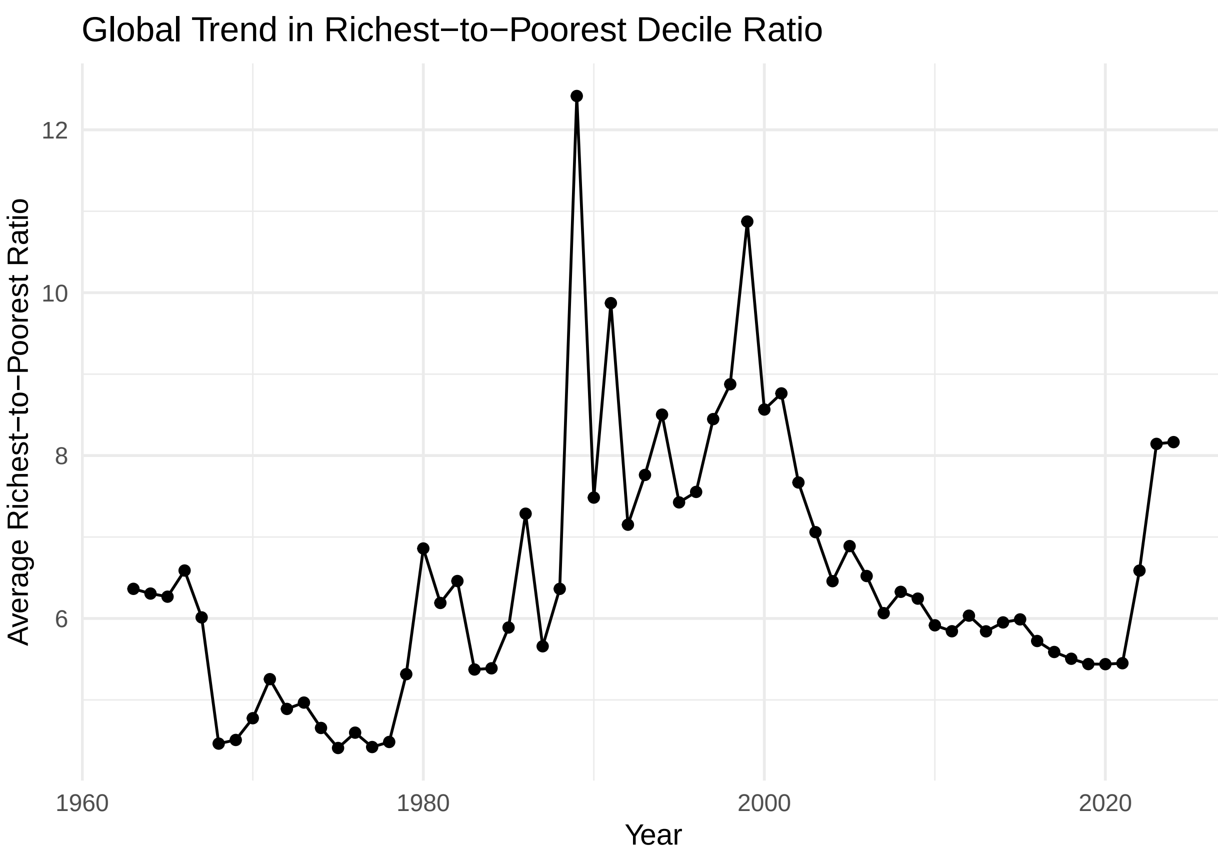
```r
head(rich_poor_trend)
```

```
## # A tibble: 6 x 4
##    Year Average Median Count
##   <int>   <dbl>  <dbl> <int>
## 1  1963    6.36   6.36     1
## 2  1964    6.31   6.31     1
## 3  1965    6.27   6.27     1
## 4  1966    6.59   6.59     1
## 5  1967    6.01   6.01     1
## 6  1968    4.46   4.46     2
```

```r
# plotting the trend of mean-to-median ratio
mean_median_plot = ggplot(mean_median_trend, aes(x = Year, y = Average)) +
  geom_line() +
  geom_point() +
  labs(title = "Global Trend in Mean-to-Median Income Ratio",
       x = "Year",
       y = "Average Mean-to-Median Ratio") +
  theme_minimal()

# save mean-median plot
ggsave(file.path(viz_dir, "mean_median_trend.png"),
       mean_median_plot,
       width = 10,
       height = 6,
       dpi = 300)
print(mean_median_plot)
```

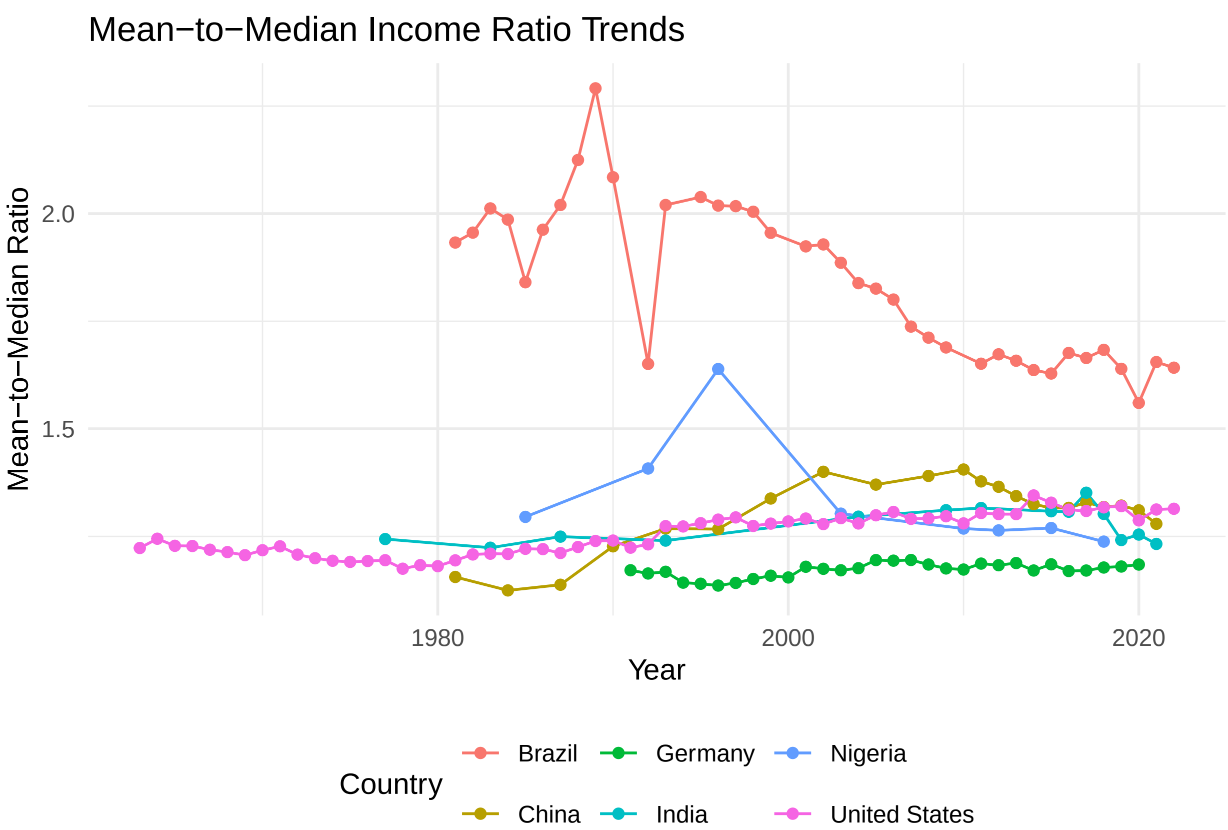



```r
# plotting the trend of rich-to-poor ratio
rich_poor_plot = ggplot(rich_poor_trend, aes(x = Year, y = Average)) +
  geom_line() +
  geom_point() +
  labs(title = "Global Trend in Richest-to-Poorest Decile Ratio",
       x = "Year",
       y = "Average Richest-to-Poorest Ratio") +
  theme_minimal()

# save rich-poor plot
ggsave(file.path(viz_dir, "rich_poor_trend.png"),
       rich_poor_plot,
       width = 10,
       height = 6,
       dpi = 300)
print(rich_poor_plot)
```

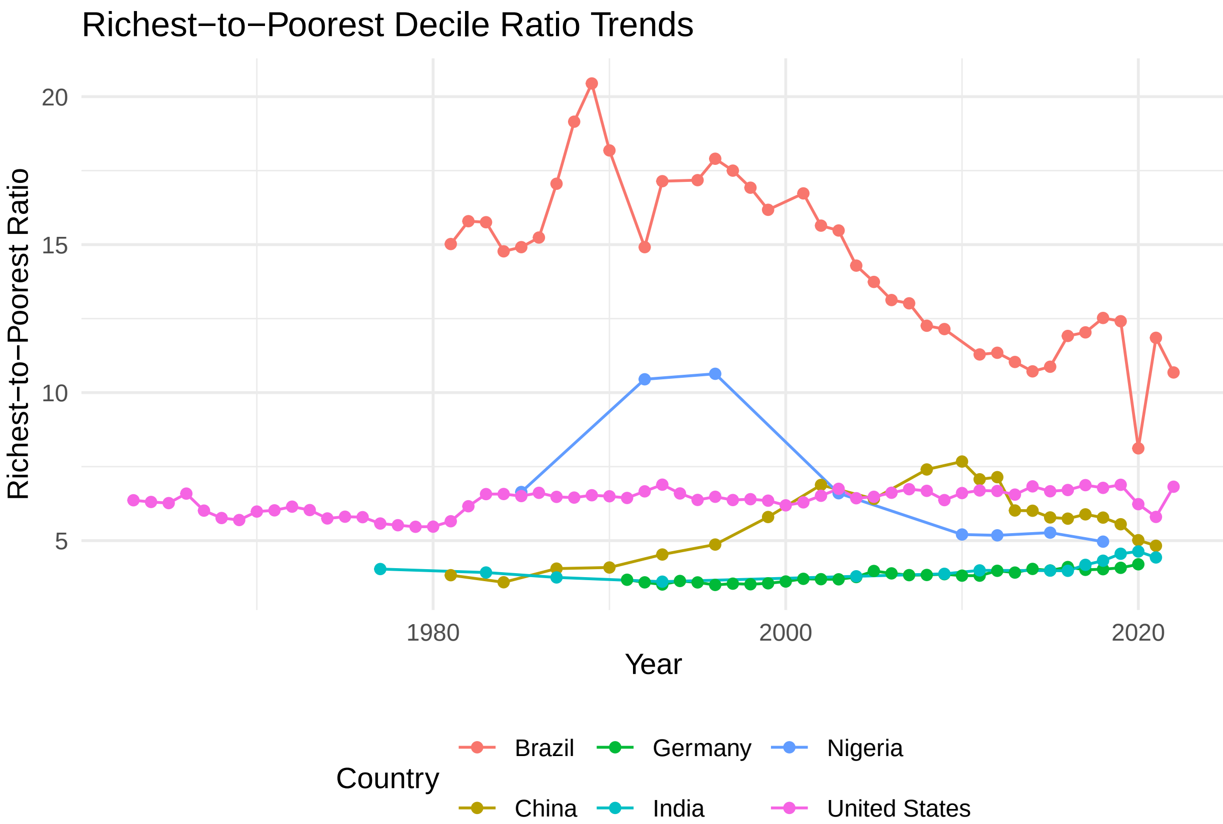



```
##########################################################################
# Part 5: Country-Specific Trends Analysis
##########################################################################

# analyzing income distribution trends for selected countries
selected_countries = c("United States", "China", "India", "Brazil", "Germany", "Nigeria")

# filtering data for selected countries
selected_data = income_distribution %>%
  filter(Country %in% selected_countries)

# plotting mean-to-median ratio trends for selected countries
country_mean_median_plot = ggplot(selected_data, aes(x = Year, y = Mean_to_Median_Ratio, color = Country
  geom_line() +
  geom_point() +
  labs(title = "Mean-to-Median Income Ratio Trends",
       x = "Year",
       y = "Mean-to-Median Ratio") +
  theme_minimal() +
  theme(legend.position = "bottom")

# save country mean-median plot
ggsave(file.path(viz_dir, "country_mean_median_trends.png"),
       country_mean_median_plot,
       width = 12,
       height = 8,
       dpi = 300)
```

```
print(country_mean_median_plot)
```




```
# plotting richest-to-poorest ratio trends for selected countries
country_rich_poor_plot = ggplot(selected_data, aes(x = Year, y = Richest_to_Poorest_Ratio, color = Coun
  geom_line() +
  geom_point() +
  labs(title = "Richest-to-Poorest Decile Ratio Trends",
       x = "Year",
       y = "Richest-to-Poorest Ratio") +
  theme_minimal() +
  theme(legend.position = "bottom")

# save country rich-poor plot
ggsave(file.path(viz_dir, "country_rich_poor_trends.png"),
       country_rich_poor_plot,
       width = 12,
       height = 8,
       dpi = 300)
print(country_rich_poor_plot)
```




```r
########################################################################
# Part 6: Poverty Analysis
########################################################################

# focusing on extreme poverty (below $2.15 a day)
# selecting just the relevant columns
extreme_poverty_share = share_below_poverty %>%
  dplyr::select(Country, Year, "Share.below..2.15.a.day")

# renaming for clarity
names(extreme_poverty_share)[3] = "Extreme_Poverty_Share"

# merging income distribution with poverty data
poverty_income = income_distribution %>%
  inner_join(extreme_poverty_share, by = c("Country", "Year"))

# checking the merged dataset
head(poverty_income)
```

```
##              Country Year Mean_Income Median_Income Poorest_Decile
## 1            Albania 2016    12.53201      10.27492       4.700507
## 2            Albania 2017    12.41122      10.18767       4.905077
## 3            Albania 2018    13.31651      11.63998       5.476367
## 4 Argentina (urban) 1980    32.47283      24.83049       8.657177
## 5 Argentina (urban) 1986    35.17031      25.36732       9.810018
## 6 Argentina (urban) 1987    32.14754      22.35962       7.710212
##   Richest_Decile Mean_to_Median_Ratio Richest_to_Poorest_Ratio Top_Bottom_Gap
```

```
## 1       23.15194           1.219669                 4.925413       18.45143
## 2       22.72124           1.218259                 4.632188       17.81616
## 3       23.32338           1.144032                 4.258914       17.84701
## 4       63.53146           1.307781                 7.338589       54.87429
## 5       69.72409           1.386442                 7.107438       59.91407
## 6       63.92279           1.437750                 8.290666       56.21258
##   Extreme_Poverty_Share
## 1              5.795102
## 2              5.264806
## 3              3.892983
## 4              0.000000
## 5              1.119132
## 6              1.186086
```

```r
dim(poverty_income)
```

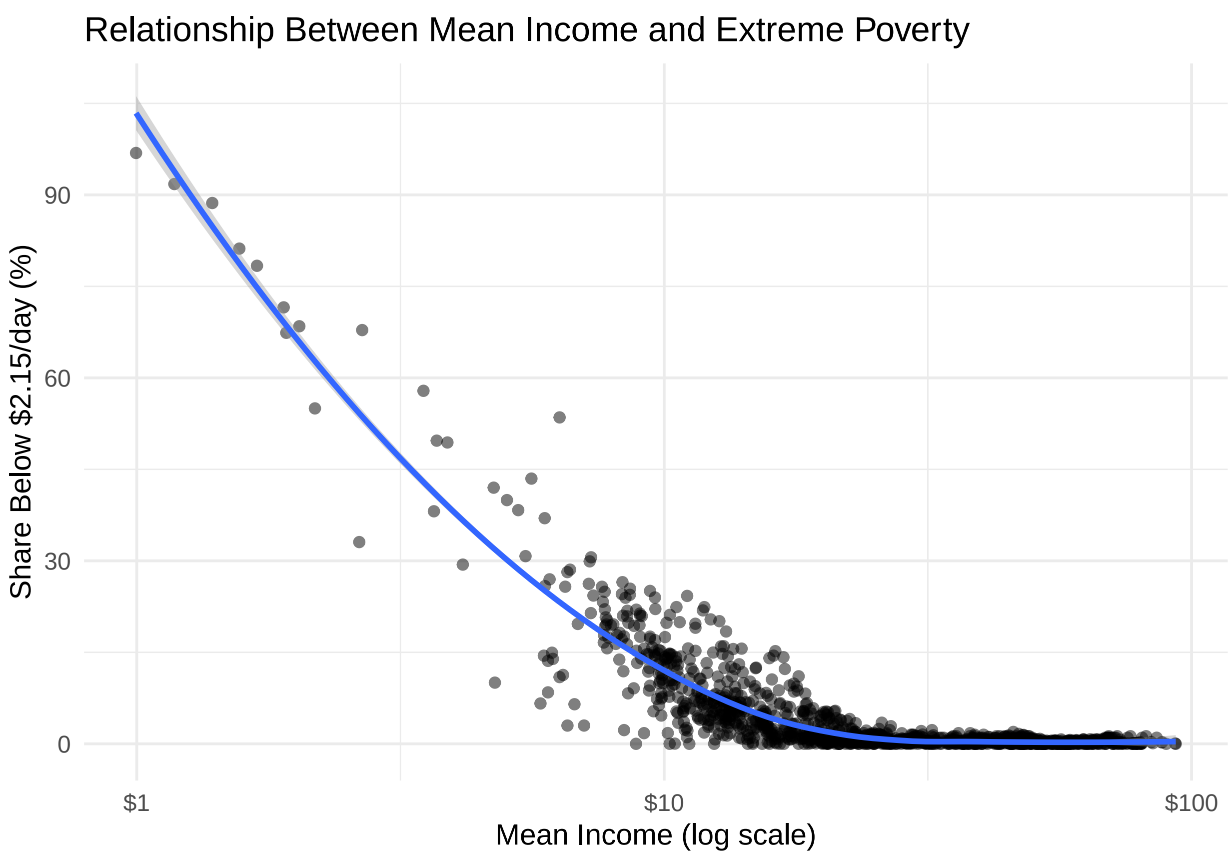
```
## [1] 1462   10
```

```r
# creating scatterplots to examine relationships
# 1. mean income vs. extreme poverty
income_poverty_plot = ggplot(poverty_income, aes(x = Mean_Income, y = Extreme_Poverty_Share)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = TRUE) +
  scale_x_log10(labels = dollar_format()) +
  labs(title = "Relationship Between Mean Income and Extreme Poverty",
       x = "Mean Income (log scale)",
       y = "Share Below $2.15/day (%)") +
  theme_minimal()

# save income-poverty plot
ggsave(file.path(viz_dir, "income_poverty_relationship.png"),
       income_poverty_plot,
       width = 10,
       height = 6,
       dpi = 300)
```

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

```r
print(income_poverty_plot)
```

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

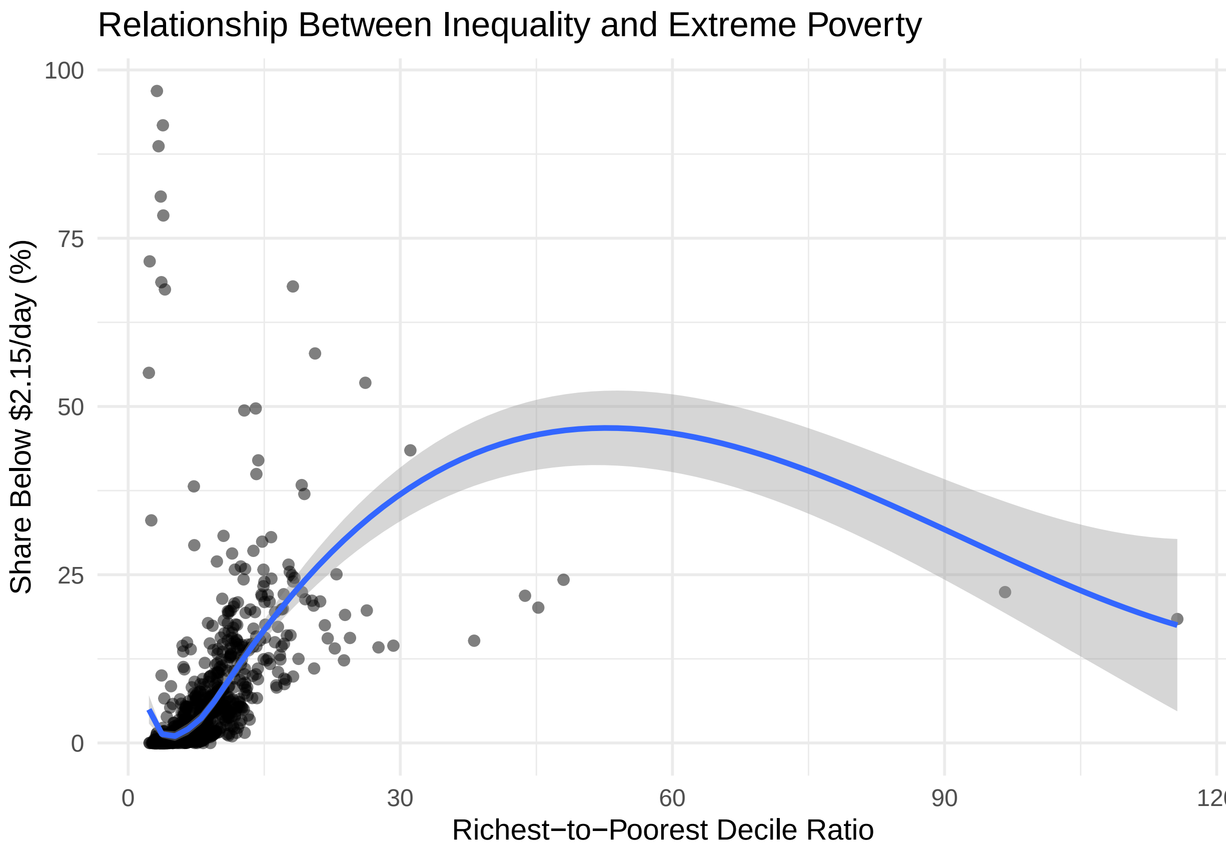



```r
# 2. inequality vs. extreme poverty
inequality_poverty_plot = ggplot(poverty_income, aes(x = Richest_to_Poorest_Ratio, y = Extreme_Poverty_S
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = TRUE) +
  labs(title = "Relationship Between Inequality and Extreme Poverty",
       x = "Richest-to-Poorest Decile Ratio",
       y = "Share Below $2.15/day (%)") +
  theme_minimal()

# save inequality-poverty plot
ggsave(file.path(viz_dir, "inequality_poverty_relationship.png"),
       inequality_poverty_plot,
       width = 10,
       height = 6,
       dpi = 300)
```

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

```r
print(inequality_poverty_plot)
```

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

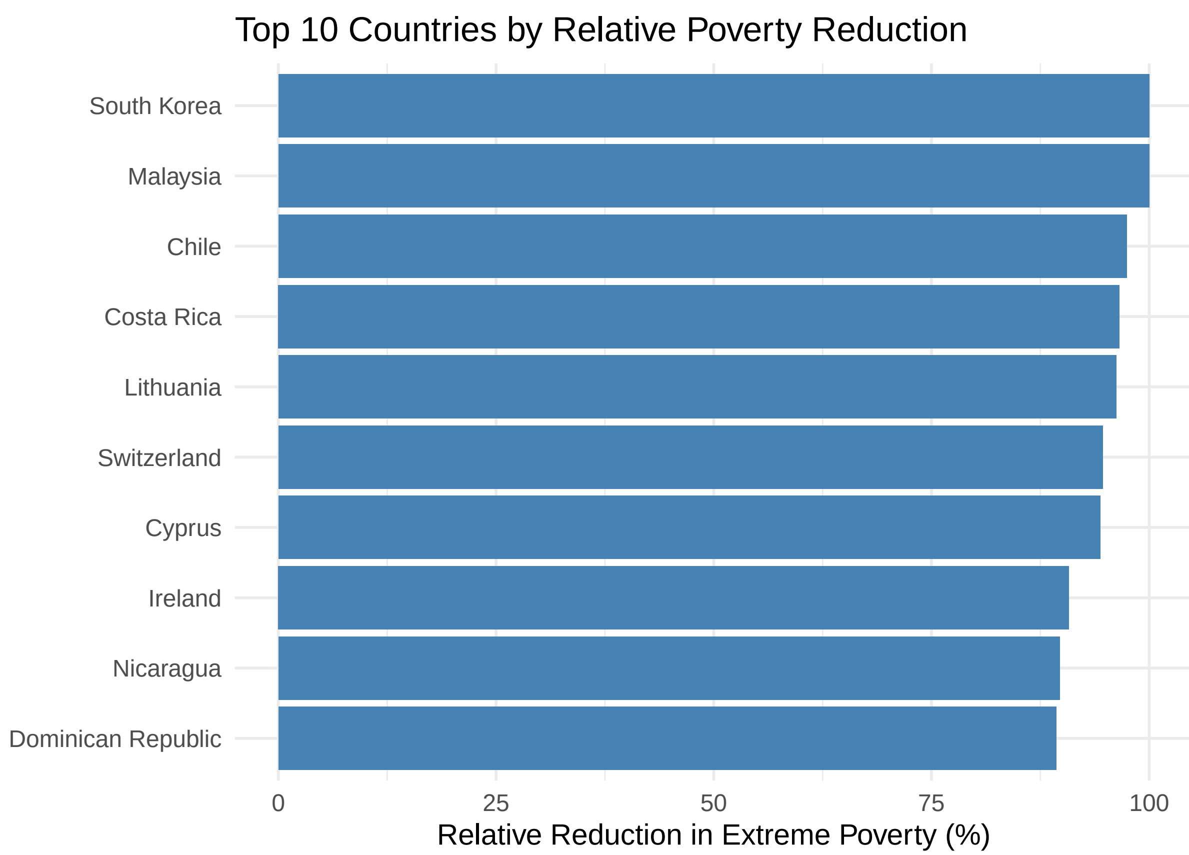



```r
######################################################################
# Part 7: Poverty Reduction Success Analysis
######################################################################

# identifying countries with substantial data over time
country_years = poverty_income %>%
  group_by(Country) %>%
  summarize(
    Years = n(),
    Min_Year = min(Year),
    Max_Year = max(Year),
    Year_Range = Max_Year - Min_Year
  ) %>%
  filter(Years >= 5, Year_Range >= 10)

# analyzing poverty reduction for countries with sufficient data
poverty_reduction = poverty_income %>%
  filter(Country %in% country_years$Country) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(
    Initial_Poverty = first(Extreme_Poverty_Share),
    Final_Poverty = last(Extreme_Poverty_Share),
    Absolute_Reduction = Initial_Poverty - Final_Poverty,
    Relative_Reduction = (Initial_Poverty - Final_Poverty) / Initial_Poverty * 100,
    Initial_Year = first(Year),
    Final_Year = last(Year),
```

```r
    Year_Range = Final_Year - Initial_Year,
    Annual_Reduction = Absolute_Reduction / Year_Range
  ) %>%
  dplyr::select(Country, Initial_Year, Final_Year, Initial_Poverty, Final_Poverty,
                Absolute_Reduction, Relative_Reduction, Annual_Reduction) %>%
  distinct(Country, .keep_all = TRUE) %>%
  arrange(desc(Relative_Reduction))

# viewing the most successful countries in poverty reduction
head(poverty_reduction, 10)
```

```
## # A tibble: 10 x 8
## # Groups:   Country [10]
##    Country            Initial_Year Final_Year Initial_Poverty Final_Poverty
##    <chr>                     <int>      <int>           <dbl>         <dbl>
##  1 Malaysia                   1984       2021           2.68          0
##  2 South Korea                2006       2021           0.249         0
##  3 Chile                      1987       2022          15.4           0.395
##  4 Costa Rica                 1981       2023          25.9           0.883
##  5 Lithuania                  1993       2021           6.62          0.251
##  6 Switzerland                1982       2020           0.697         0.0372
##  7 Cyprus                     2004       2021           0.0941        0.00530
##  8 Ireland                    1987       2021           0.689         0.0635
##  9 Nicaragua                  1993       2014          38.3           3.94
## 10 Dominican Republic         1986       2022           7.09          0.757
## # i 3 more variables: Absolute_Reduction <dbl>, Relative_Reduction <dbl>,
## #   Annual_Reduction <dbl>
```

```r
# plotting top 10 countries by relative poverty reduction
top_reducers = head(poverty_reduction, 10)

poverty_reduction_plot = ggplot(top_reducers, aes(x = reorder(Country, Relative_Reduction), y = Relativ
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 Countries by Relative Poverty Reduction",
       x = "",
       y = "Relative Reduction in Extreme Poverty (%)") +
  theme_minimal()

# save poverty reduction plot
ggsave(file.path(viz_dir, "top_poverty_reducers.png"),
       poverty_reduction_plot,
       width = 10,
       height = 6,
       dpi = 300)
print(poverty_reduction_plot)
```




```
#####################################################################
# Part 8: Statistical Inference And Hypothesis Testing
#####################################################################

# hypothesis testing for income inequality
# h0: mean-to-median ratio = 1 (perfect equality)
# h1: mean-to-median ratio > 1 (inequality exists)
t_test_inequality = t.test(income_distribution$Mean_to_Median_Ratio, mu = 1, alternative = "greater")
print(t_test_inequality)
```

```
##
##  One Sample t-test
##
## data:  income_distribution$Mean_to_Median_Ratio
## t = 61.688, df = 2634, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 1
## 95 percent confidence interval:
##  1.349036      Inf
## sample estimates:
## mean of x
##  1.358601
```

```
# creating regional groupings first
poverty_income = poverty_income %>%
  mutate(Region = case_when(
    Country %in% c("China", "Japan", "South Korea", "Vietnam") ~ "East Asia",
```

```r
    Country %in% c("India", "Pakistan", "Bangladesh", "Sri Lanka") ~ "South Asia",
    Country %in% c("Nigeria", "South Africa", "Kenya", "Ethiopia") ~ "Sub-Saharan Africa",
    Country %in% c("Brazil", "Mexico", "Argentina", "Colombia") ~ "Latin America",
    Country %in% c("Germany", "France", "United Kingdom", "Italy") ~ "Europe",
    TRUE ~ "Other"
  ))

# additional hypothesis tests
# test for difference in poverty rates between regions
region_test = aov(Extreme_Poverty_Share ~ Region, data = poverty_income)
summary(region_test)
```

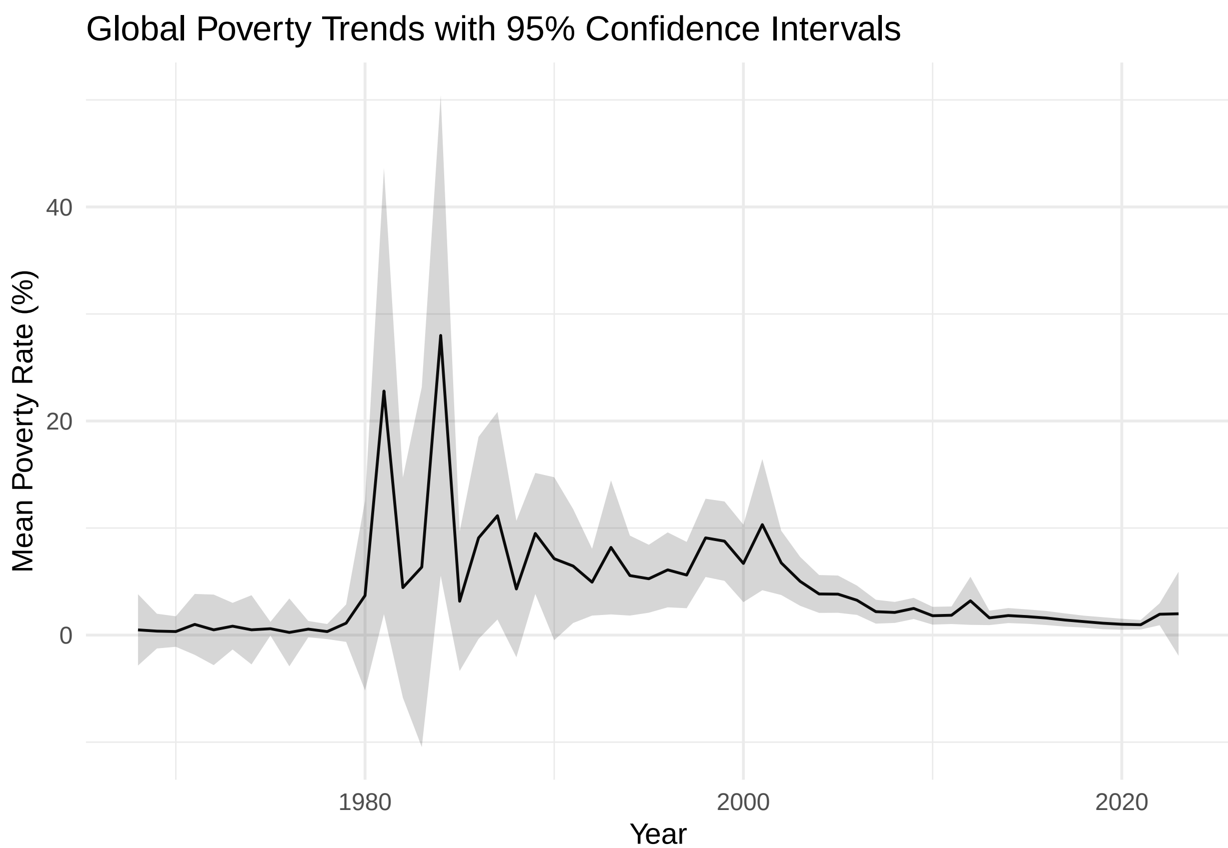
```
##               Df Sum Sq Mean Sq F value Pr(>F)
## Region         3   7494  2498.1   33.27 <2e-16 ***
## Residuals   1458 109464    75.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# test for correlation between income and poverty
cor_test = cor.test(poverty_income$Mean_Income, poverty_income$Extreme_Poverty_Share)
print(cor_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  poverty_income$Mean_Income and poverty_income$Extreme_Poverty_Share
## t = -20.977, df = 1460, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5196850 -0.4408487
## sample estimates:
##        cor
## -0.4812393
```

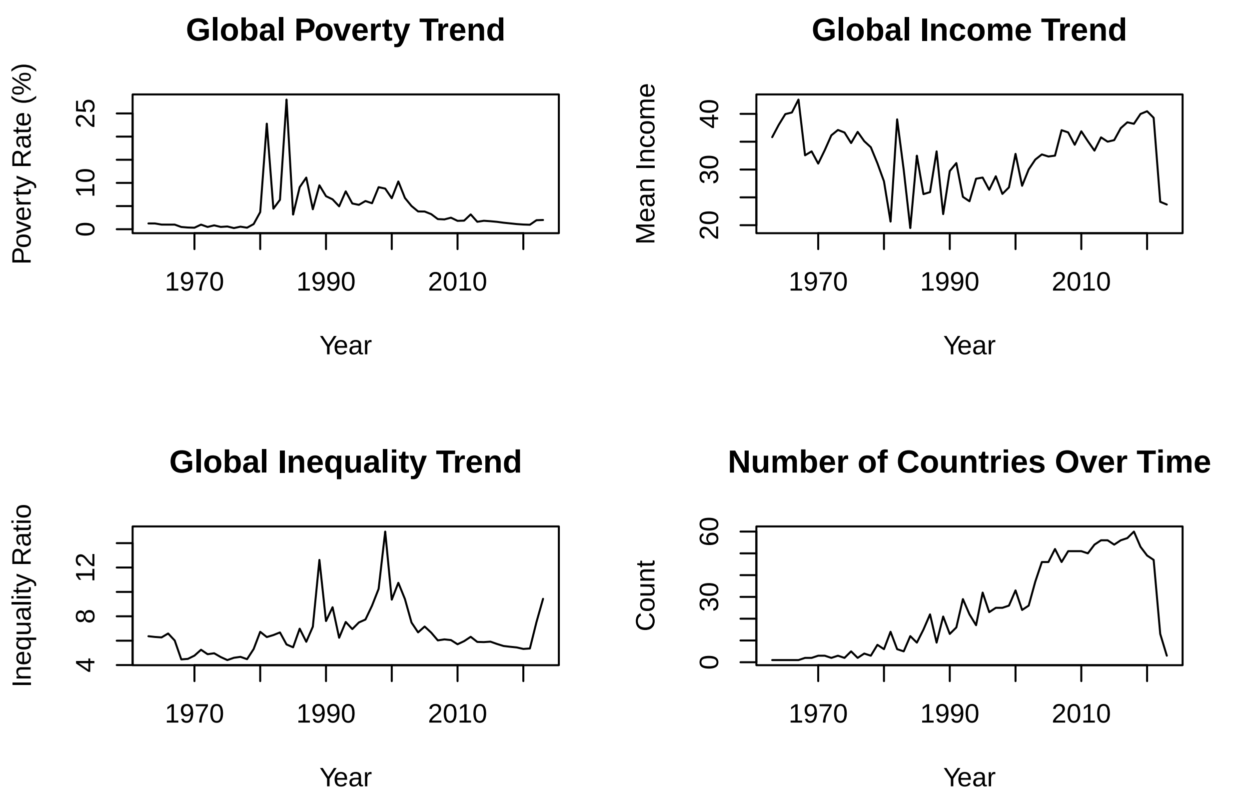
```r
# confidence intervals for poverty reduction
poverty_ci = poverty_income %>%
  group_by(Year) %>%
  filter(n() > 1) %>%
  summarize(
    mean_poverty = mean(Extreme_Poverty_Share, na.rm = TRUE),
    se = sd(Extreme_Poverty_Share, na.rm = TRUE) / sqrt(n()),
    ci_lower = ifelse(n() > 1, mean_poverty - qt(0.975, n()-1) * se, mean_poverty),
    ci_upper = ifelse(n() > 1, mean_poverty + qt(0.975, n()-1) * se, mean_poverty)
  )

# plotting poverty trends with confidence intervals
poverty_ci_plot = ggplot(poverty_ci, aes(x = Year, y = mean_poverty)) +
  geom_line() +
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper), alpha = 0.2) +
  labs(title = "Global Poverty Trends with 95% Confidence Intervals",
       x = "Year",
       y = "Mean Poverty Rate (%)") +
```

```r
  theme_minimal()

# save poverty ci plot
ggsave(file.path(viz_dir, "poverty_trends_with_ci.png"),
       poverty_ci_plot,
       width = 10,
       height = 6,
       dpi = 300)
print(poverty_ci_plot)
```




```r
######################################################################
# Part 9: Time Series Analysis
######################################################################

# step 1: data preparation and initial visualization
# calculate global averages over time
global_trends = poverty_income %>%
  group_by(Year) %>%
  summarize(
    mean_poverty = mean(Extreme_Poverty_Share, na.rm = TRUE),
    mean_income = mean(Mean_Income, na.rm = TRUE),
    mean_inequality = mean(Richest_to_Poorest_Ratio, na.rm = TRUE),
    n_countries = n()
  )

# visualize global trends
```

```r
par(mfrow = c(2, 2))
plot(global_trends$Year, global_trends$mean_poverty, type = "l",
     main = "Global Poverty Trend", xlab = "Year", ylab = "Poverty Rate (%)")
plot(global_trends$Year, global_trends$mean_income, type = "l",
     main = "Global Income Trend", xlab = "Year", ylab = "Mean Income")
plot(global_trends$Year, global_trends$mean_inequality, type = "l",
     main = "Global Inequality Trend", xlab = "Year", ylab = "Inequality Ratio")
plot(global_trends$Year, global_trends$n_countries, type = "l",
     main = "Number of Countries Over Time", xlab = "Year", ylab = "Count")
```

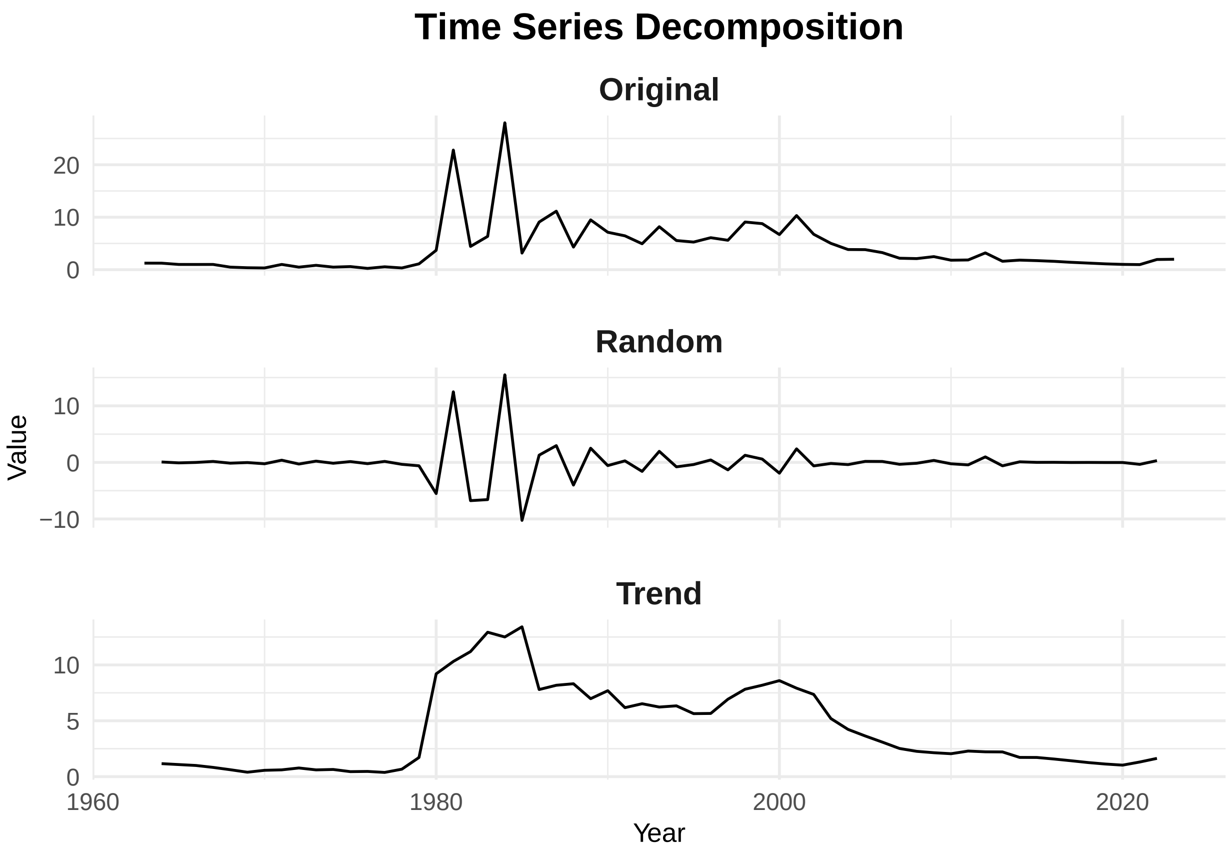



```r
par(mfrow = c(1, 1))

# step 2: time series decomposition
# ensure we have enough data points for decomposition
if(nrow(global_trends) >= 2) {
  # create time series object with appropriate frequency
  # for annual data, we'll use frequency = 1
  poverty_ts = ts(global_trends$mean_poverty,
                  start = min(global_trends$Year),
                  end = max(global_trends$Year),
                  frequency = 1)

  # perform decomposition using a moving average approach
  # calculate trend using moving average
  trend = ma(poverty_ts, order = 3, centre = TRUE)

  # calculate detrended series
  detrended = poverty_ts - trend
```

```r
  # prepare data for ggplot
  decomposition_data = data.frame(
    Year = global_trends$Year,
    Original = as.numeric(poverty_ts),
    Trend = as.numeric(trend),
    Random = as.numeric(detrended)
  )

  # create long format data for plotting
  decomposition_long = decomposition_data %>%
    pivot_longer(
      cols = c(Original, Trend, Random),
      names_to = "Component",
      values_to = "Value"
    )

  # create decomposition plot using ggplot2
  decomposition_plot = ggplot(decomposition_long, aes(x = Year, y = Value)) +
    geom_line() +
    facet_wrap(~ Component, ncol = 1, scales = "free_y") +
    labs(title = "Time Series Decomposition",
         x = "Year",
         y = "Value") +
    theme_minimal() +
    theme(
      strip.text = element_text(size = 12, face = "bold"),
      plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
      axis.title = element_text(size = 10),
      panel.spacing = unit(1, "lines")
    )

  # print the plot
  print(decomposition_plot)

  # calculate and print trend statistics
  trend_stats = data.frame(
    Component = c("Original", "Trend", "Random"),
    Mean = c(mean(poverty_ts, na.rm = TRUE),
             mean(trend, na.rm = TRUE),
             mean(detrended, na.rm = TRUE)),
    SD = c(sd(poverty_ts, na.rm = TRUE),
           sd(trend, na.rm = TRUE),
           sd(detrended, na.rm = TRUE))
  )

  print("Time Series Decomposition Statistics:")
  print(trend_stats)

} else {
  cat("Insufficient data points for time series decomposition.\n")
}
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
```

```
## ('geom_line()').
```




```
## [1] "Time Series Decomposition Statistics:"
##   Component          Mean       SD
## 1  Original  4.1264018296 4.942975
## 2     Trend  4.2116677518 3.715937
## 3    Random -0.0002787988 3.418257
```

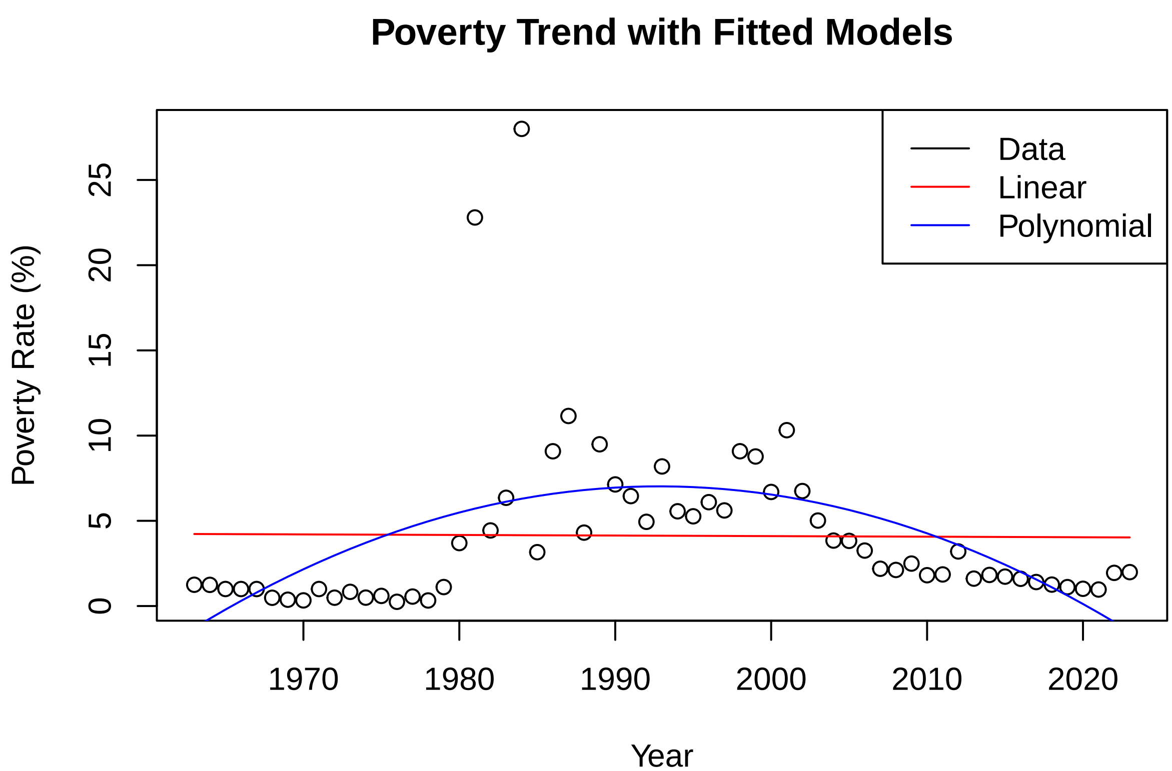
```
# step 3: trend analysis
# fit linear and polynomial models to poverty trend
linear_trend = lm(mean_poverty ~ Year, data = global_trends)
poly_trend = lm(mean_poverty ~ poly(Year, 2), data = global_trends)

# compare models
trend_comparison = data.frame(
  model = c("Linear", "Polynomial"),
  aic = c(AIC(linear_trend), AIC(poly_trend)),
  adj_r_squared = c(summary(linear_trend)$adj.r.squared,
                    summary(poly_trend)$adj.r.squared)
)
print("Trend Model Comparison:")
```

```
## [1] "Trend Model Comparison:"
```

```r
print(trend_comparison)
```

```
##       model      aic adj_r_squared
## 1    Linear 373.0454   -0.01680197
## 2 Polynomial 355.0828    0.25434962
```

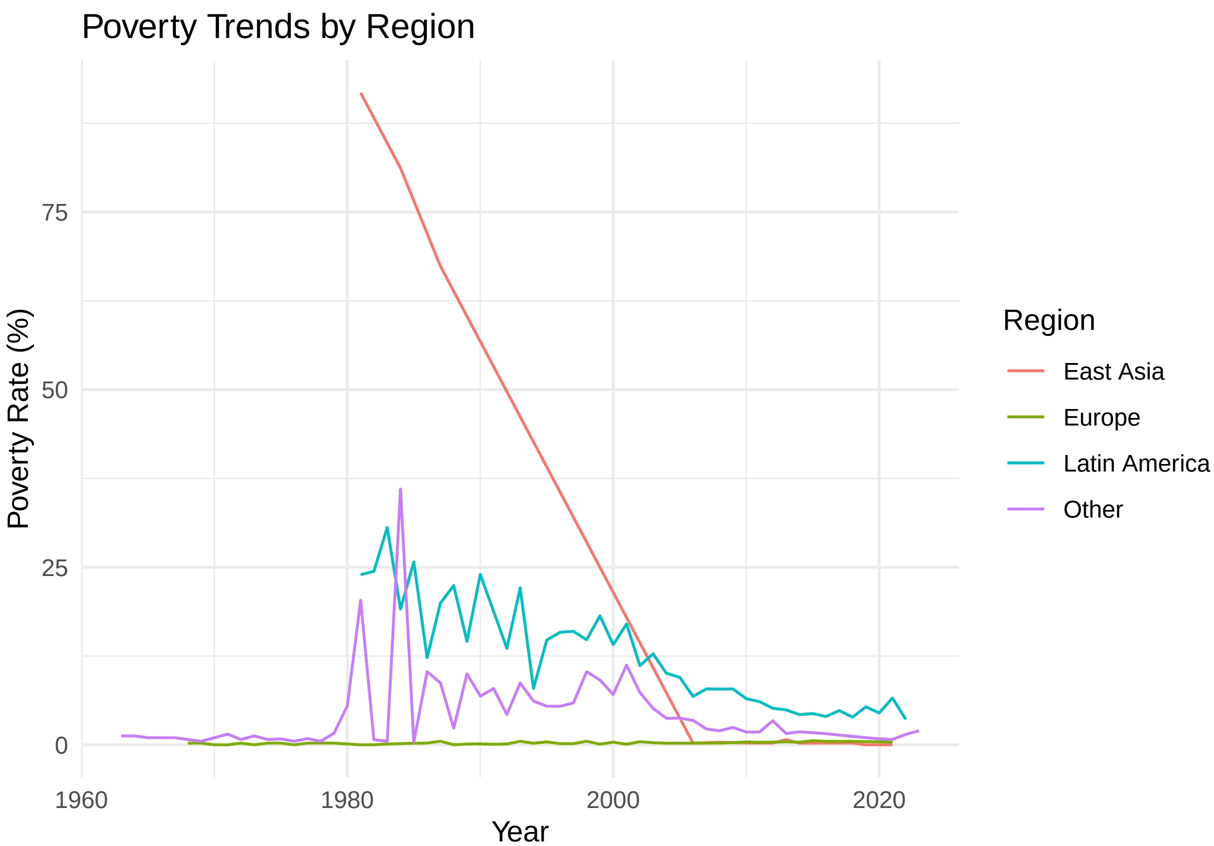
```r
# visualize trend fits
plot(global_trends$Year, global_trends$mean_poverty,
     main = "Poverty Trend with Fitted Models",
     xlab = "Year", ylab = "Poverty Rate (%)")
lines(global_trends$Year, predict(linear_trend), col = "red")
lines(global_trends$Year, predict(poly_trend), col = "blue")
legend("topright", legend = c("Data", "Linear", "Polynomial"),
       col = c("black", "red", "blue"), lty = 1)
```




```r
# step 4: regional analysis
# calculate regional trends
regional_trends = poverty_income %>%
  group_by(Region, Year) %>%
  summarize(
    mean_poverty = mean(Extreme_Poverty_Share, na.rm = TRUE),
    mean_income = mean(Mean_Income, na.rm = TRUE),
    mean_inequality = mean(Richest_to_Poorest_Ratio, na.rm = TRUE),
    n_countries = n()
  )
```

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

```r
# visualize regional trends
ggplot(regional_trends, aes(x = Year, y = mean_poverty, color = Region)) +
  geom_line() +
  labs(title = "Poverty Trends by Region",
       x = "Year",
       y = "Poverty Rate (%)") +
  theme_minimal()
```




```r
# step 5: statistical tests for regional differences
# anova test for regional differences
regional_anova = aov(Extreme_Poverty_Share ~ Region, data = poverty_income)
summary(regional_anova)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## Region         3   7494  2498.1   33.27 <2e-16 ***
## Residuals   1458 109464    75.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

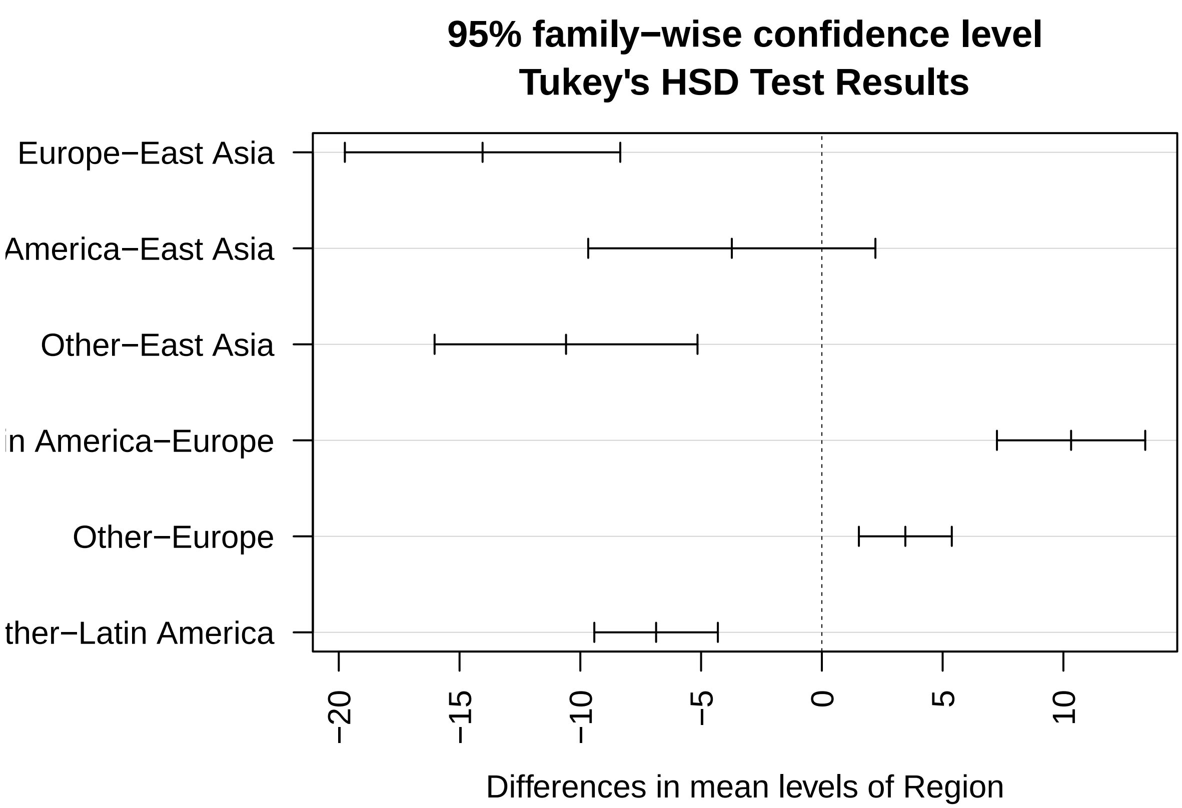
```r
# tukey's hsd test for pairwise comparisons
tukey_test = TukeyHSD(regional_anova)
print("Regional Pairwise Comparisons:")
```

```
## [1] "Regional Pairwise Comparisons:"
```

```r
print(tukey_test)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Extreme_Poverty_Share ~ Region, data = poverty_income)
##
## $Region
##                                diff        lwr       upr     p adj
## Europe-East Asia         -14.047315 -19.748552 -8.346077 0.0000000
## Latin America-East Asia   -3.728336  -9.673632  2.216960 0.3715701
## Other-East Asia          -10.591331 -16.034165 -5.148496 0.0000037
## Latin America-Europe      10.318979   7.249671 13.388287 0.0000000
## Other-Europe               3.455984   1.532823  5.379146 0.0000245
## Other-Latin America       -6.862995  -9.420530 -4.305459 0.0000000
```

```r
# visualize tukey's hsd results
par(mar = c(5, 8, 4, 2))  # adjust margins for better label visibility
plot(tukey_test, las = 2)
title("Tukey's HSD Test Results", line = 1)
```




```r
par(mar = c(5, 4, 4, 2))  # reset margins to default

# step 6: trend strength analysis
# install and load trend package if not already installed
if (!require("trend")) {
  install.packages("trend")
```

```r
  library(trend)
}
```

```
## Loading required package: trend
```

```r
# calculate trend strength using mann-kendall test
mk_test = mk.test(global_trends$mean_poverty)
print("Mann-Kendall Trend Test:")
```

```
## [1] "Mann-Kendall Trend Test:"
```

```r
print(mk_test)
```

```
##
##  Mann-Kendall trend test
##
## data:  global_trends$mean_poverty
## z = 0.57873, n = 61, p-value = 0.5628
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 9.400000e+01 2.582333e+04 5.136612e-02
```

```r
# calculate trend magnitude using sen's slope
sen_slope = sens.slope(global_trends$mean_poverty)
print("Sen's Slope Estimate:")
```

```
## [1] "Sen's Slope Estimate:"
```

```r
print(sen_slope)
```

```
##
##  Sen's slope
##
## data:  global_trends$mean_poverty
## z = 0.57873, n = 61, p-value = 0.5628
## alternative hypothesis: true z is not equal to 0
## 95 percent confidence interval:
##  -0.05945888  0.02914683
## sample estimates:
## Sen's slope
## 0.009001064
```

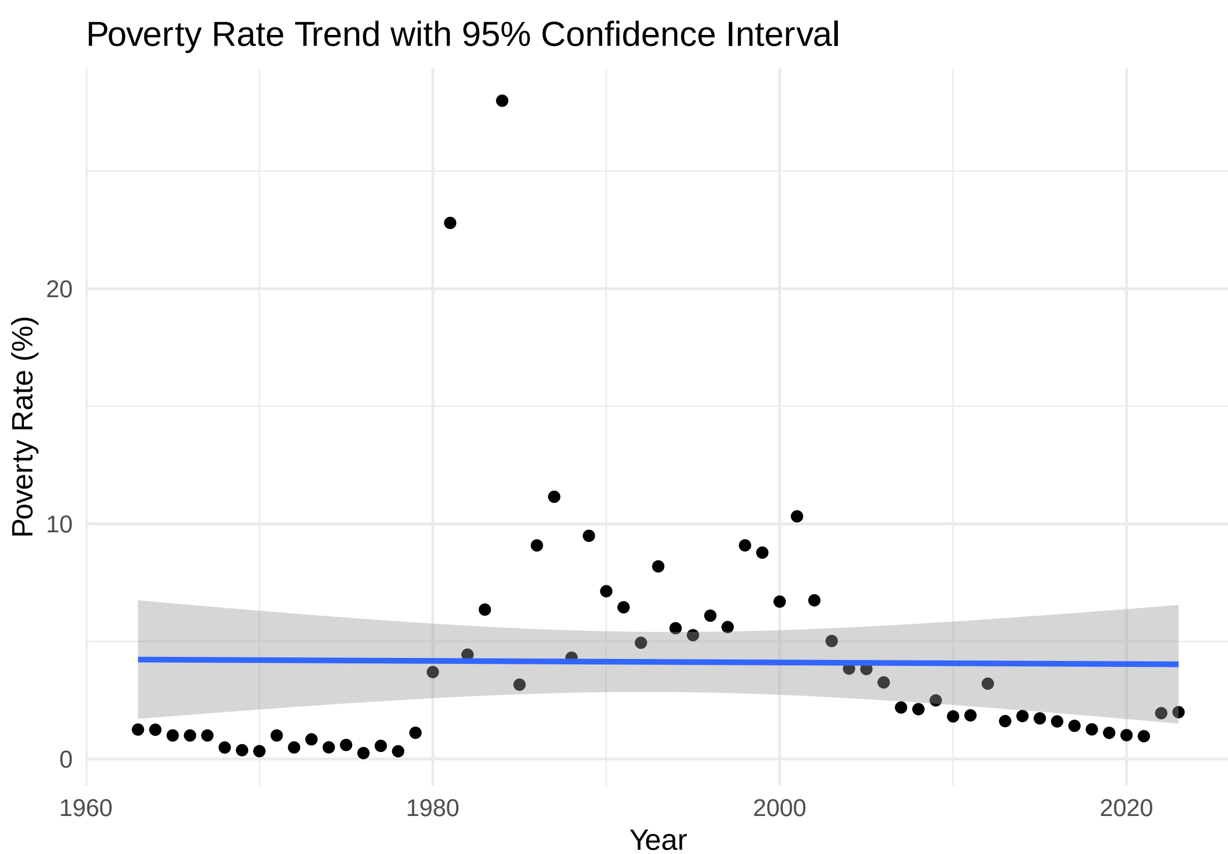
```r
# calculate trend strength metrics
trend_strength = list(
  mk_test = mk_test,
  sen_slope = sen_slope,
  direction = ifelse(sen_slope$estimates < 0, "Decreasing", "Increasing"),
  magnitude = abs(sen_slope$estimates),
  significance = ifelse(mk_test$p.value < 0.05, "Significant", "Not Significant")
```

```
)

# visualize trend with confidence intervals
trend_plot = ggplot(global_trends, aes(x = Year, y = mean_poverty)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Poverty Rate Trend with 95% Confidence Interval",
       x = "Year",
       y = "Poverty Rate (%)") +
  theme_minimal()

# save trend plot
ggsave(file.path(viz_dir, "poverty_trend.png"),
       trend_plot,
       width = 10,
       height = 6,
       dpi = 300)
```

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

```
print(trend_plot)
```

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



```r
# step 7: forecasting
# fit arima model for forecasting
library(forecast)
poverty_arima = auto.arima(poverty_ts)
print("ARIMA Model Summary:")
```

```
## [1] "ARIMA Model Summary:"
```

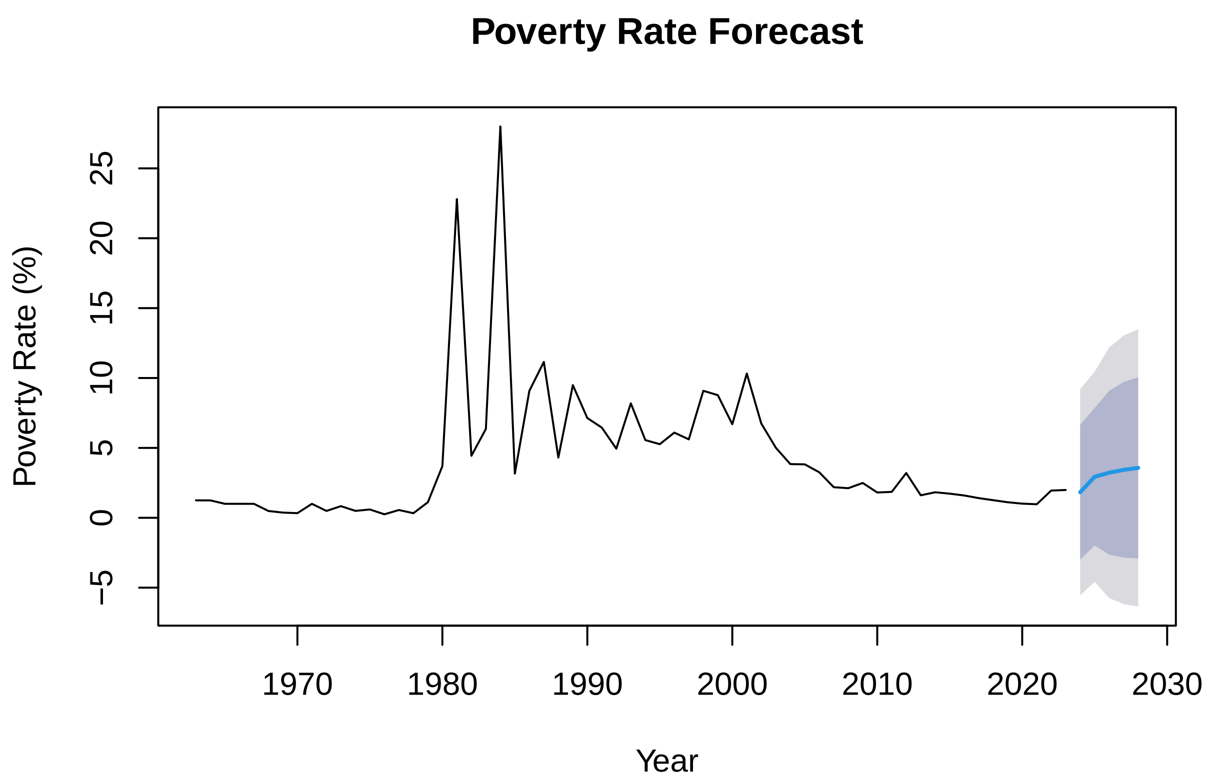
```r
print(summary(poverty_arima))
```

```
## Series: poverty_ts
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ma1     ma2    mean
##       0.7088  -0.9113  0.8056  3.9242
## s.e.  0.1117   0.0868  0.0938  1.3651
##
## sigma^2 = 14.14:  log likelihood = -166.56
## AIC=343.11   AICc=344.2   BIC=353.66
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE    MAPE      MASE
## Training set 0.05509096 3.635338 1.976413 -72.31039 94.4935 0.7965445
##                    ACF1
## Training set 0.06930241
```

```r
# generate forecasts
forecast_poverty = forecast(poverty_arima, h = 5)

# save forecast plot
png(file.path(viz_dir, "poverty_forecast.png"),
    width = 10,
    height = 6,
    units = "in",
    res = 300)
plot(forecast_poverty,
     main = "Poverty Rate Forecast",
     xlab = "Year",
     ylab = "Poverty Rate (%)")
dev.off()
```

```
## pdf
##   2
```

```r
# print forecast plot
plot(forecast_poverty,
     main = "Poverty Rate Forecast",
     xlab = "Year",
     ylab = "Poverty Rate (%)")
```




```
# step 8: results summary
cat("\nTime Series Analysis Results:\n")
```

```
##
## Time Series Analysis Results:
```

```
cat("1. Trend Analysis:\n")
```

```
## 1. Trend Analysis:
```

```
cat("   - Best Trend Model:",
    trend_comparison$model[which.min(trend_comparison$aic)], "\n")
```

```
##    - Best Trend Model: Polynomial
```

```
cat("   - Trend Direction:",
    ifelse(coef(linear_trend)[2] < 0, "Decreasing", "Increasing"), "\n")
```

```
##    - Trend Direction: Decreasing
```

```
cat("   - Trend Significance:",
    ifelse(summary(linear_trend)$coefficients[2,4] < 0.05, "Significant", "Not Significant"), "\n\n")
```

```
##    - Trend Significance: Not Significant
```

```r
cat("2. Regional Analysis:\n")
```

```
## 2. Regional Analysis:
```

```r
cat("   - Regional Differences:",
    ifelse(summary(regional_anova)[[1]][1,5] < 0.05, "Significant", "Not Significant"), "\n")
```

```
##    - Regional Differences: Significant
```

```r
cat("   - Number of Significant Pairwise Differences:",
    sum(tukey_test$Region[,4] < 0.05), "\n\n")
```

```
##    - Number of Significant Pairwise Differences: 5
```

```r
cat("3. Trend Strength:\n")
```

```
## 3. Trend Strength:
```

```r
cat("   - Mann-Kendall Test:",
    ifelse(trend_strength$mk_test$p.value < 0.05, "Significant Trend", "No Significant Trend"), "\n")
```

```
##    - Mann-Kendall Test: No Significant Trend
```

```r
cat("   - Sen's Slope:", round(trend_strength$sen_slope$estimates, 3), "Units Per Year\n")
```

```
##    - Sen's Slope: 0.009 Units Per Year
```

```r
cat("   - Trend Direction:", trend_strength$direction, "\n")
```

```
##    - Trend Direction: Increasing
```

```r
cat("   - Trend Magnitude:", round(trend_strength$magnitude, 3), "Units Per Year\n\n")
```

```
##    - Trend Magnitude: 0.009 Units Per Year
```

```r
cat("4. Forecasting:\n")
```

```
## 4. Forecasting:
```

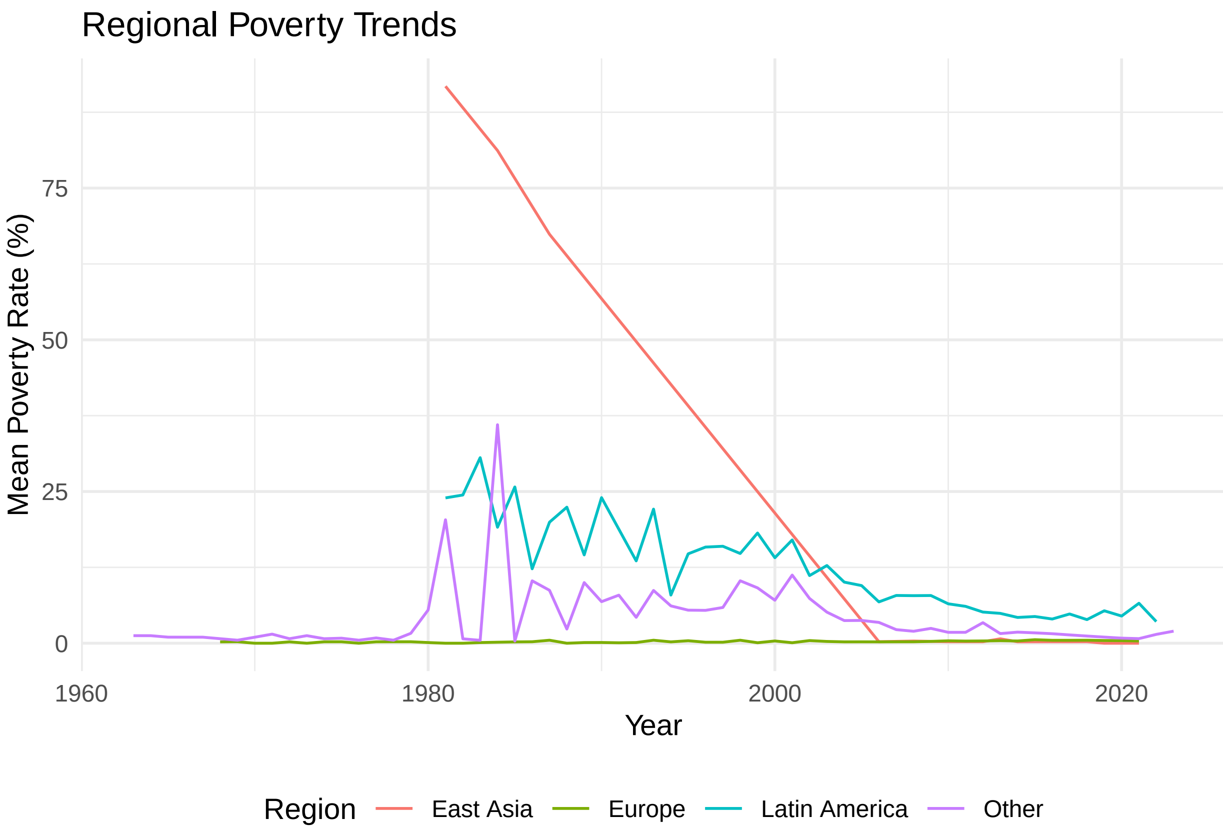
```r
cat("   - Best ARIMA Model:", poverty_arima$arma[1], ",", poverty_arima$arma[2], "\n")
```

```
##    - Best ARIMA Model: 1 , 2
```

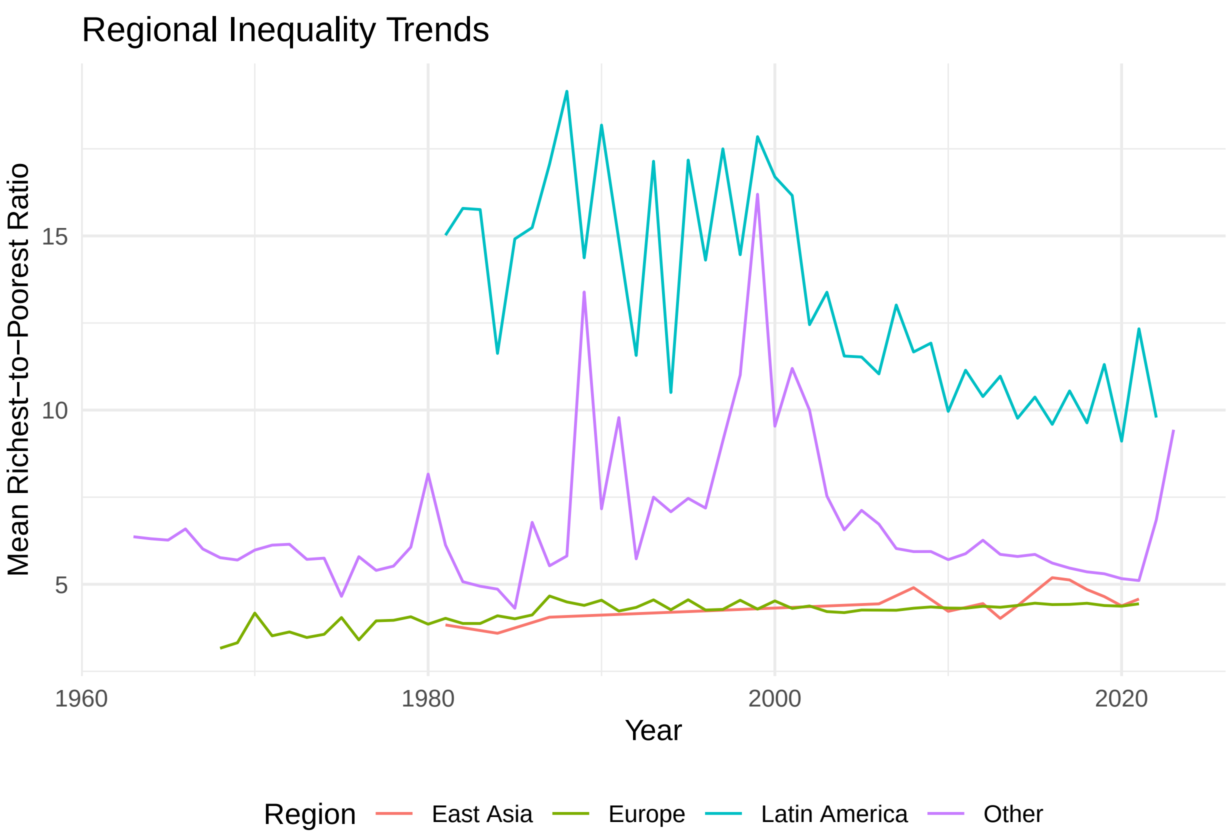
```r
cat("   - Forecast Accuracy (MAPE):",
    round(accuracy(poverty_arima)[5], 2), "%\n")
```

```
##    - Forecast Accuracy (MAPE): 94.49 %
```

```r
######################################################################
# Part 10: Regional And Comparative Analysis
######################################################################

# calculating regional statistics
regional_stats = poverty_income %>%
  group_by(Region, Year) %>%
  summarize(
    mean_poverty = mean(Extreme_Poverty_Share, na.rm = TRUE),
    median_poverty = median(Extreme_Poverty_Share, na.rm = TRUE),
    mean_inequality = mean(Richest_to_Poorest_Ratio, na.rm = TRUE),
    median_inequality = median(Richest_to_Poorest_Ratio, na.rm = TRUE),
    .groups = "drop"
  )

# plotting regional trends
regional_trends_plot = ggplot(regional_stats, aes(x = Year, y = mean_poverty, color = Region)) +
  geom_line() +
  labs(title = "Regional Poverty Trends",
       x = "Year",
       y = "Mean Poverty Rate (%)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# save regional poverty trends plot
ggsave(file.path(viz_dir, "regional_poverty_trends.png"),
       regional_trends_plot,
       width = 12,
       height = 8,
       dpi = 300)
print(regional_trends_plot)
```

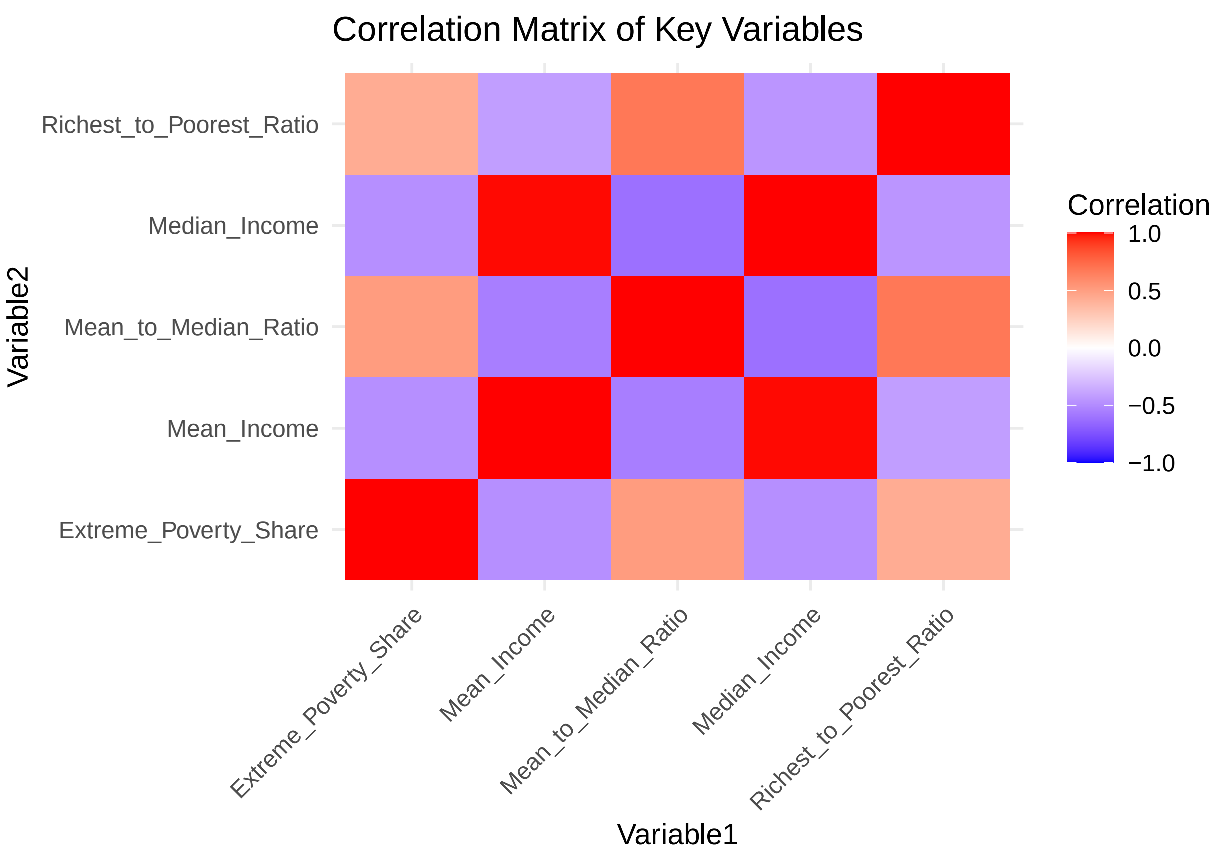



```r
# regional inequality trends
regional_inequality_plot = ggplot(regional_stats, aes(x = Year, y = mean_inequality, color = Region)) +
  geom_line() +
  labs(title = "Regional Inequality Trends",
       x = "Year",
       y = "Mean Richest-to-Poorest Ratio") +
  theme_minimal() +
  theme(legend.position = "bottom")

# save regional inequality trends plot
ggsave(file.path(viz_dir, "regional_inequality_trends.png"),
       regional_inequality_plot,
       width = 12,
       height = 8,
       dpi = 300)
print(regional_inequality_plot)
```




```
########################################################################
# Part 11: Advanced Statistical Analysis
########################################################################

# calculating correlation matrix with all relevant variables
correlation_matrix = poverty_income %>%
  dplyr::select(Mean_Income, Median_Income, Richest_to_Poorest_Ratio,
         Mean_to_Median_Ratio, Extreme_Poverty_Share) %>%
  cor(use = "complete.obs")

# creating correlation heatmap
correlation_long = as.data.frame(correlation_matrix) %>%
  rownames_to_column("Variable1") %>%
  pivot_longer(-Variable1, names_to = "Variable2", values_to = "Correlation")

advanced_correlation_plot = ggplot(correlation_long, aes(x = Variable1, y = Variable2, fill = Correlati
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                      midpoint = 0, limit = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Matrix of Key Variables")

# save advanced correlation matrix plot
ggsave(file.path(viz_dir, "advanced_correlation_matrix.png"),
       advanced_correlation_plot,
       width = 10,
```

```r
       height = 8,
       dpi = 300)
print(advanced_correlation_plot)
```




```r
# multiple regression with interaction terms
advanced_model = lm(Extreme_Poverty_Share ~ log(Mean_Income) * Richest_to_Poorest_Ratio +
                    Mean_to_Median_Ratio, data = poverty_income)
summary(advanced_model)
```

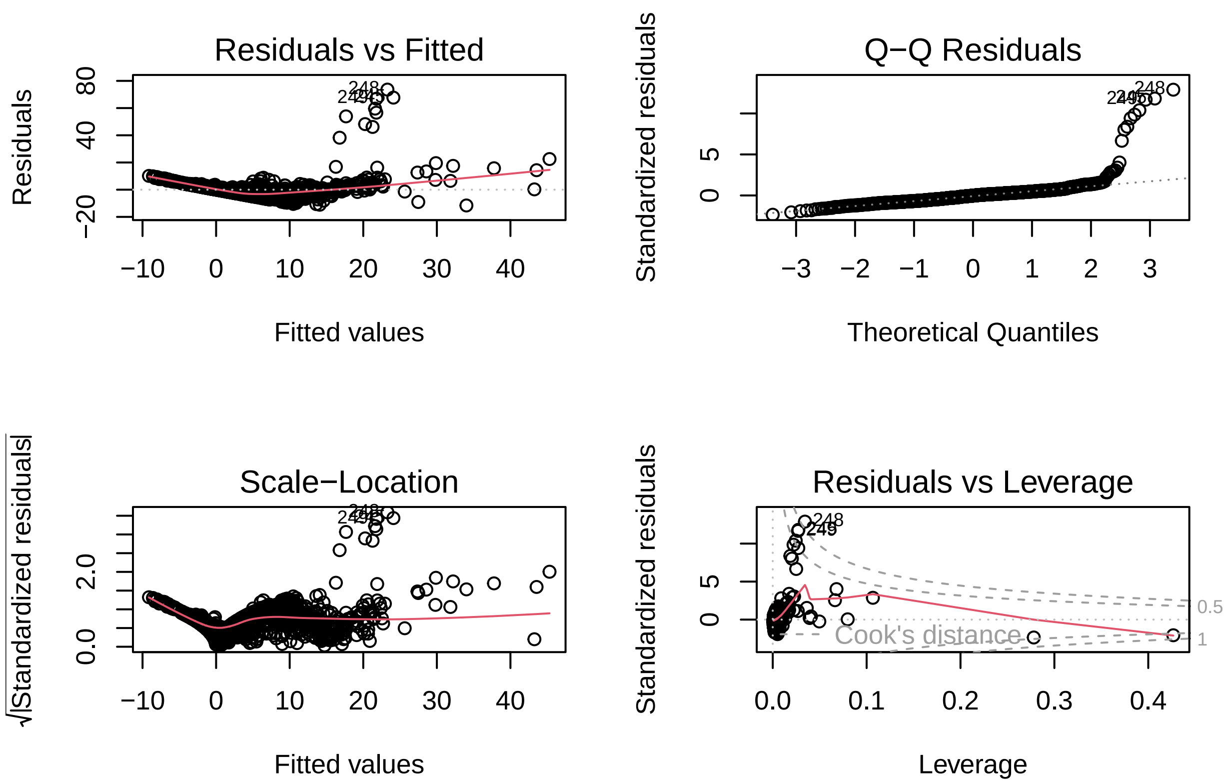
```
##
## Call:
## lm(formula = Extreme_Poverty_Share ~ log(Mean_Income) * Richest_to_Poorest_Ratio +
##     Mean_to_Median_Ratio, data = poverty_income)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.609  -2.865  -0.130   1.851  73.588
##
## Coefficients:
##                                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                12.08126    2.07444   5.824 7.06e-09
## log(Mean_Income)                           -2.87520    0.44391  -6.477 1.28e-10
## Richest_to_Poorest_Ratio                    2.69230    0.18345  14.676  < 2e-16
## Mean_to_Median_Ratio                        2.41441    1.03327   2.337   0.0196
## log(Mean_Income):Richest_to_Poorest_Ratio -0.98273    0.07024 -13.990  < 2e-16
##
```

```
## (Intercept)                             ***
## log(Mean_Income)                        ***
## Richest_to_Poorest_Ratio                ***
## Mean_to_Median_Ratio                    *
## log(Mean_Income):Richest_to_Poorest_Ratio ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.808 on 1457 degrees of freedom
## Multiple R-squared:  0.5797, Adjusted R-squared:  0.5786
## F-statistic: 502.5 on 4 and 1457 DF,  p-value: < 2.2e-16
```

```r
# save diagnostic plots for advanced model
png(file.path(viz_dir, "advanced_model_diagnostics.png"),
    width = 10,
    height = 8,
    units = "in",
    res = 300)
par(mfrow = c(2, 2))
plot(advanced_model)
par(mfrow = c(1, 1))
dev.off()
```

```
## pdf
##   2
```

```r
# display diagnostic plots in rstudio
par(mfrow = c(2, 2))
plot(advanced_model)
```



```r
par(mfrow = c(1, 1))

######################################################################
# Part 12: Results Export And Summary
######################################################################

# creating comprehensive summary statistics
summary_stats = poverty_income %>%
  group_by(Year) %>%
  summarize(
    mean_poverty = mean(Extreme_Poverty_Share, na.rm = TRUE),
    median_poverty = median(Extreme_Poverty_Share, na.rm = TRUE),
    sd_poverty = sd(Extreme_Poverty_Share, na.rm = TRUE),
    mean_inequality = mean(Richest_to_Poorest_Ratio, na.rm = TRUE),
    median_inequality = median(Richest_to_Poorest_Ratio, na.rm = TRUE),
    sd_inequality = sd(Richest_to_Poorest_Ratio, na.rm = TRUE)
  )

# saving all results
write.csv(summary_stats, file.path(output_dir, "summary_statistics.csv"), row.names = FALSE)
write.csv(correlation_matrix, file.path(output_dir, "correlation_matrix.csv"), row.names = TRUE)
write.csv(regional_stats, file.path(output_dir, "regional_analysis.csv"), row.names = FALSE)

# printing final summary
cat("\nAnalysis Complete!\n")
```

```
##
## Analysis Complete!
```

```r
cat("Results have been saved to the output directory.\n")
```

```
## Results have been saved to the output directory.
```

```r
cat("Visualizations have been saved to the output/visualizations directory.\n")
```

```
## Visualizations have been saved to the output/visualizations directory.
```

```r
cat("Key findings:\n")
```

```
## Key findings:
```

```r
cat("1. Global poverty trends show",
    ifelse(tail(summary_stats$mean_poverty, 1) < head(summary_stats$mean_poverty, 1),
           "a decreasing trend", "an increasing trend"),
    "over the study period.\n")
```

```
## 1. Global poverty trends show an increasing trend over the study period.
```

```
cat("2. The correlation between inequality and poverty is",
    round(correlation_matrix["Richest_to_Poorest_Ratio", "Extreme_Poverty_Share"], 3), "\n")
```

```
## 2. The correlation between inequality and poverty is 0.428
```

```
cat("3. Regional analysis shows",
    regional_stats$Region[which.min(regional_stats$mean_poverty[regional_stats$Year == max(regional_sta
    "has the lowest poverty rates in the most recent year.\n")
```

```
## 3. Regional analysis shows East Asia has the lowest poverty rates in the most recent year.
```

```
cat("4. Time series decomposition indicates",
    trend_strength$direction, "trend with",
    ifelse(trend_strength$significance == "Significant", "significant", "non-significant"),
    "change.\n")
```

```
## 4. Time series decomposition indicates Increasing trend with non-significant change.
```

```
cat("5. Advanced regression analysis shows",
    ifelse(summary(advanced_model)$coefficients["log(Mean_Income):Richest_to_Poorest_Ratio", "Pr(>|t|)"]
           "a significant interaction between income and inequality",
           "no significant interaction between income and inequality"), ".\n")
```

```
## 5. Advanced regression analysis shows a significant interaction between income and inequality .
```

```
# create a function to save base R plots
save_base_plot = function(plot_func, filename, width = 10, height = 6) {
  png(file.path(viz_dir, filename),
      width = width,
      height = height,
      units = "in",
      res = 300)
  plot_func()
  dev.off()
}

# save global trends plots
save_base_plot(
  function() {
    par(mfrow = c(2, 2))
    plot(global_trends$Year, global_trends$mean_poverty, type = "l",
         main = "Global Poverty Trend", xlab = "Year", ylab = "Poverty Rate (%)")
    plot(global_trends$Year, global_trends$mean_income, type = "l",
         main = "Global Income Trend", xlab = "Year", ylab = "Mean Income")
    plot(global_trends$Year, global_trends$mean_inequality, type = "l",
         main = "Global Inequality Trend", xlab = "Year", ylab = "Inequality Ratio")
    plot(global_trends$Year, global_trends$n_countries, type = "l",
         main = "Number of Countries Over Time", xlab = "Year", ylab = "Count")
    par(mfrow = c(1, 1))
  },
```

```r
  "global_trends.png",
  width = 12,
  height = 10
)
```

```
## pdf
##   2
```

```r
# save trend fits plot
save_base_plot(
  function() {
    plot(global_trends$Year, global_trends$mean_poverty,
         main = "Poverty Trend with Fitted Models",
         xlab = "Year", ylab = "Poverty Rate (%)")
    lines(global_trends$Year, predict(linear_trend), col = "red")
    lines(global_trends$Year, predict(poly_trend), col = "blue")
    legend("topright", legend = c("Data", "Linear", "Polynomial"),
           col = c("black", "red", "blue"), lty = 1)
  },
  "trend_fits.png"
)
```

```
## pdf
##   2
```

```r
# save tukey's hsd plot
save_base_plot(
  function() {
    par(mar = c(5, 8, 4, 2))
    plot(tukey_test, las = 2)
    title("Tukey's HSD Test Results", line = 1)
    par(mar = c(5, 4, 4, 2))
  },
  "tukey_hsd_results.png",
  width = 10,
  height = 8
)
```

```
## pdf
##   2
```

```r
# (removed redundant save of regional_poverty_trends.png; already created in part 10)

# (removed redundant save of regional_inequality_trends.png; already created in part 10)

# (removed redundant save of advanced_correlation_matrix.png; already created in part 11)
```