# Coding Sample

Chaeyeong

2025-01-06

## Descriptions

### Introduction (Motivation and Research Question)

This is the writeup summarizing the final project for Data and Programming for Public Policy II - R Programming. The research is motivated by the global issue and development programs; Persistent poverty in Low-income countries and trends of finance-focused poverty alleviation programs. Despite a global decline in poverty rates over recent 20 years, low-income countries have remained nearly stagnant, trapped into poverty where over 50% of the population continues to struggle for less than $2.15 per day (World Bank. (2024). Poverty and Inequality Platform. World Bank Group. https://pip.worldbank.org/). To address this poverty issue, one of the most popular poverty alleviation program is microfinance, targetting minorities who are inaccessible to financial services and provide complements. This raised curiosity about how financial accessibility can improve poverty such that the research question below is suggested: How and what type of financial inclusion for the poor is correlated with poverty levels across countries?

### Data Source

The research involves five data sources:

1. World Bank Global Findex Database 2021: 4 financial inclusion indicators (account, saving, borrowing, and digital payment) for income, poorest 40% (% ages 15+) across countries in 2014, 2017, and 2021 (directly downloaded its .xlsx from the world bank homepage: https://www.worldbank.org/en/publication/globalfindex/Data, choosing country-level .xlsx file.)
2. World Development Indicators: Poverty headcount ratio (%) (used the package "WDI" in R)
3. IMF Real GDP Growth (retrieved from the API as the .json version from IMF DataMapper API documentation: https://www.imf.org/external/datamapper/api/v1/NGDP_RPCH)
4. Natural Earth: world choropleth map (directly downloaded its .zip file, including the shape version from the natural earth homepage: https://www.naturalearthdata.com/downloads/10m-cultural-vectors/, choosing Admin o - Countries (version 5.1.1))
5. CAN MICROFINANCE UNLOCK A POVERTY TRAP FOR SOME ENTREPRENEURS?: The research paper (directly downloaded its pdf file from the NATIONAL BUREAU of ECONOMIC RESEARCH homepage: https://www.nber.org/papers/w26346)

You can see the detailed data processing in data.R.

## Methodology

This study employs a quantitative, correlational research method with panel data mainly sourced from World Bank. It is hypothesized that there is a negative correlation between financial inclusion percentages for the poor and poverty headcount ratio. Due to year variation across countries, the analysis uses panel data across 47 countries among a total of 89 countries, which have all years (2014, 2017 , and 2021) data. A Fixed-Effects panel regression model controls unobserved time-invariant characteristics at the country level, with clustered standard errors to account for correlation within countries over time. This design enables to apply fixed-effect for a balanced panel, which provides robust insights into the temporal relationships between financial inclusion variables and poverty across years and controls for country-level unobserved heterogenity. However, as the sample size was reduced, it potentially limits the generalizability of results. Data wrangling, visualization by static plots and shiny app, text processing, and model fitting are primary code approaches taken for this project. For further detailed code, please refer to each R script and README file.

## Research Design

The research is designed for Fixed-Effects panel regression model. Its formula is as follows:

$\text{Poverty\_Headcount\_Ratio}_{it} = \beta 0 + \beta 1(Account_Ownership_it) + \beta 2(Savings_it) + \beta 3(Borrowing_it) + \beta 4(Digital_Payments_it) + \beta 5(GDP_it) + \alpha_i + \gamma_t + \varepsilon_{it}$

Dependent Variable: Poverty Level: Poverty_Headcount_Ratio

Independent Variables (Financial Inclusion): Account_Onership: Percentage of account ownership among income, poorest 40% (% ages 15+). Savings: Percentage of those saved any money, among income, poorest 40% (% ages 15+). Borrowing: Percentage of those borrowed any money among income, poorest 40% (% ages 15+). Digital_Payments: Percentage of those Made or received a digital payment among income, poorest 40% (% ages 15+).

Control Variables: Real GDP Growth

$i$: Country. $t$: Year. $\alpha_i$: Country fixed effects. $\gamma_t$: Year fixed effects. $\epsilon_{it}$: Error term

## Visualization

There are two static plots and shinyapps, respectively. "plot_financial_inclusion_by_poverty.png" shows the estimated financial inclusion for the poor from the linear model with the poverty headcount ratio, differentiating each financial inclusion indicators. Except for borrowing, most indicators show a negative relationship with poverty. "choropleth_financial_inclusion.png" provides an overview of average financial inclusion extent for the poor in the world by scatter plots for each country in the world map; more blue scatter plot is, higher average financial inclusion is. Please see the detailed coding for those two static plots in staticplot.R

In shiny apps, the first app shows scatter plots allowing to select a country and year and visualize how financial inclusion for the poor correlates with poverty. The second app is text process with the pdf file "Microfinance_poverty_trap.pdf", the research "CAN MICROFINANCE UNLOCK A POVERTY TRAP FOR SOME ENTREPRENEURS?". The app was created for line graphs showing keywords specific sentiment. The keywords were selected for the most 10 common words used in the paper. The app allows users to select one over 10 keywords, and shows its sentiment trend across page numbers, which is estimated by AFINN. The first page and reference pages are omitted in the graph. This app helps readers identify contexts for reading the paper that how each keyword is used and refer to specific page number by showing keyword's sentiment across page numbers.

### Conclusion and implications

| Variable | Coefficient | Standard Error |
|---|---|---|
| Account Ownership (Income, poorest 40%) | 0.505 | (3.344) |
| Savings (Income, poorest 40%) | 0.757 | (1.200) |
| Borrowing (Income, poorest 40%) | 3.524* | (1.807) |
| Digital Payments (Income, poorest 40%) | -3.146 | (2.595) |
| Real GDP Growth | -0.004 | (0.034) |
| **Observations** | 141 | |
| **$R^2$** | 0.172 | |
| **Adjusted $R^2$** | -0.302 | |
| **F Statistic** | 3.703*** (df = 5; 89) | |
| **Note** | *p<0.1;* ***p<0.05;*** p<0.01 | |

The regression results show that a 10 percentage point increase in borrowing among the 40% poorest is associated with an increase of 0.3524 percentage points in the poverty rate at 10% significance level in 47 countries in 2014, 2017, and 2021. This indicates hypothesis fails; the borrowing is rather positively associated with poverty and drived further research on potential issues with debt management among the poorest groups. For further research, please see the text processing in the second shiny app above in visualization section. Meanwhile, coefficients for other indicators are not statistically significant, reflecting data limitation or limited direct effects on poverty rates in the studied context. Please see the detailed coding for model fitting in model.R.

# Codes

## Setup

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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 error
```

```
library(httr)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
```

```
## The following object is masked from 'package:purrr':
##
##     flatten
```

```r
library(WDI)
library(readxl)
library(sf)
```

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

```r
library(pdftools)
```

```
## Using poppler version 24.11.0
```

```r
library(tidytext)
library(udpipe)
library(igraph)
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
##
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
##
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
##
## The following object is masked from 'package:tidyr':
##
##     crossing
##
## The following object is masked from 'package:tibble':
##
##     as_data_frame
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

```r
library(ggraph)
library(shiny)
```

```
##
## Attaching package: 'shiny'
##
## The following object is masked from 'package:jsonlite':
##
##     validate
```

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:igraph':
##
##     groups
##
## The following object is masked from 'package:httr':
##
##     config
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(plm)
```

```
##
## Attaching package: 'plm'
##
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```r
library(stargazer)
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
library(lmtest)
```

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

```r
library(styler)
```

```
##
## Attaching package: 'styler'
##
## The following object is masked from 'package:httr':
##
##     cache_info
```

```r
style_file("coding sample.Rmd")
```

```
## Styling  1  files:
##  coding sample.Rmd v
## ----------------------------------------
## Status   Count   Legend
## v    1   File unchanged.
## i    0   File changed.
## x    0   Styling threw an error.
## ----------------------------------------
```

```r
setwd("/Users/yeong/Documents/GitHub/R-II_final/")
```

## Data loading and cleaning

```r
# Load and retrieve raw data
## Define data folder path
data_path <- "data/"

## Load Global Findex 2021 Database
path_global_findex <- paste0(data_path, "DatabankWide.xlsx")
global_findex_raw <- read_xlsx(path_global_findex,
  sheet = "Data"
)

## Retrieve World Development Indicators from package (Extract poverty headcount ratio)
retrieve_wdi <- function(indicator = "SI.POV.DDAY", start_year = 2014, end_year = 2021) {
  raw_file <- paste0(data_path, "wdi_raw.csv")

  if (!file.exists(raw_file)) {
    filtered <- WDI(indicator = indicator, start = start_year, end = end_year, extra = TRUE)
    write_csv(filtered, raw_file)
    wdi_raw <- read_csv(raw_file)
  } else {
    wdi_raw <- read_csv(raw_file)
  }
  return(wdi_raw)
}

wdi_raw <- retrieve_wdi()
```

```
## Rows: 2128 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): country, iso2c, iso3c, region, capital, income, lending
## dbl  (4): year, SI.POV.DDAY, longitude, latitude
## lgl  (1): status
## date (1): lastupdated
##
## 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.
```

```
## Retrieve IMF's GDP from API
retrieve_raw_json <- function() {
  raw_file <- paste0(data_path, "imf_gdp_growth_raw.json")

  if (!file.exists(raw_file)) {
    url <- "https://www.imf.org/external/datamapper/api/v1/NGDP_RPCH"
    response <- GET(url)
    writeLines(content(response, as = "text", encoding = "UTF-8"), raw_file)
    gdp_raw <- read_json(raw_file)
  } else {
    gdp_raw <- read_json(raw_file)
  }
  return(gdp_raw)
}

gdp_raw <- retrieve_raw_json()

## Load the downloaded a shapefile for the world map (used for shinyapp afterwords)
zipF <- paste0(data_path, "ne_10m_admin_0_countries.zip")
unzip(zipF, exdir = data_path)

# Data Cleaning
## Global Findex
financial_inclusion_poor <- c(
  "Account, income, poorest 40% (% ages 15+)",
  "Saved any money, income, poorest 40% (% ages 15+)",
  "Borrowed any money, income, poorest 40% (% ages 15+)",
  "Made or received a digital payment, income, poorest 40% (% ages 15+)"
)

global_findex_clean <- global_findex_raw %>%
  select(`Country name`, `Country code`, Year, `Income group`, all_of(financial_inclusion_poor)) %>%
  na.omit() %>%
  rename(
    per_account_ownership_poor = `Account, income, poorest 40% (% ages 15+)`,
    per_saving_poor = `Saved any money, income, poorest 40% (% ages 15+)`,
    per_borrowing_poor = `Borrowed any money, income, poorest 40% (% ages 15+)`,
    per_digital_payment_poor = `Made or received a digital payment, income, poorest 40% (% ages 15+)`
  ) %>%
  rename_all(tolower)

## World Development Indicator
wdi_clean <- wdi_raw %>%
  select(country, year, income, SI.POV.DDAY, latitude, longitude) %>%
  filter(!income %in% c("Aggregates")) %>%
  na.omit() %>%
  rename(poverty_headcount_ratio = SI.POV.DDAY) %>%
  rename_all(tolower)

## GDP from IMF (converting to csv file)
gdp_df <- as.data.frame(gdp_raw)
gdp_df_longer <- gdp_df %>%
  select(where(is.numeric)) %>%
  pivot_longer(
    cols = everything(),
    names_to = c("type", "indicator", "country code", "year"),
    names_sep = "\\."
  ) %>%
  rename(gdp = value)

gdp_clean <- gdp_df_longer %>%
  select(`country code`, year, gdp) %>%
  filter(year >= 2014) %>%
  na.omit()

# Merge those datasets
combined <- global_findex_clean %>%
  inner_join(wdi_clean, by = c(
    `country name` = "country",
    "year",
    `income group` = "income"
  ))

gdp_clean$year <- as.numeric(gdp_clean$year)

combined <- combined %>%
  inner_join(gdp_clean, by = c("country code", "year"))

# Reshape the combined dataset longer and write it for further plotting
inclusive_poverty_gdp_long <- combined %>%
  pivot_longer(cols = starts_with("per_"), names_to = "metric", values_to = "value") %>%
  group_by(`country name`) %>%
  mutate(avg_value = mean(value, na.rm = TRUE))

write_csv(inclusive_poverty_gdp_long, paste0(data_path, "inclusive_poverty_gdp_long.csv"))

# Further exploratory data analysis for the research design
```

```r
## Create a table to look at yearly country counts for the research design
## to decide either using panel data or cross sectional data
total_countries <- n_distinct(combined$`country name`)

all_years_countries <- combined %>%
  group_by(`country name`) %>%
  summarise(year_count = n_distinct(year)) %>%
  filter(year_count == 3) %>%
  nrow()

two_years_countries <- combined %>%
  group_by(`country name`) %>%
  summarise(year_count = n_distinct(year)) %>%
  filter(year_count == 2) %>%
  nrow()

one_year_countries <- combined %>%
  group_by(`country name`) %>%
  summarise(year_count = n_distinct(year)) %>%
  filter(year_count == 1) %>%
  nrow()

yearly_country_count <- combined %>%
  group_by(year) %>%
  summarise(
    yearly_countries = n_distinct(`country name`)
  ) %>%
  mutate(
    total_countries = total_countries,
    countries_missing_years = total_countries - yearly_countries,
    countries_all_years = all_years_countries,
    countries_two_years_only = two_years_countries,
    countries_one_year_only = one_year_countries
  )
view(yearly_country_count)

# Prepare data for the model fitting (fixed-effect panel regression model)
## Filter countries having 3 years
three_years_only <- combined %>%
  group_by(`country name`) %>%
  filter(n_distinct(year) == 3)

## Write the filtered dataset
write_csv(three_years_only, paste0(data_path, "inclusive_poverty_gdp_model.csv"))
```

## Static Plot

```r
inclusive_poverty_gdp_long <- read_csv("data/inclusive_poverty_gdp_long.csv")
```

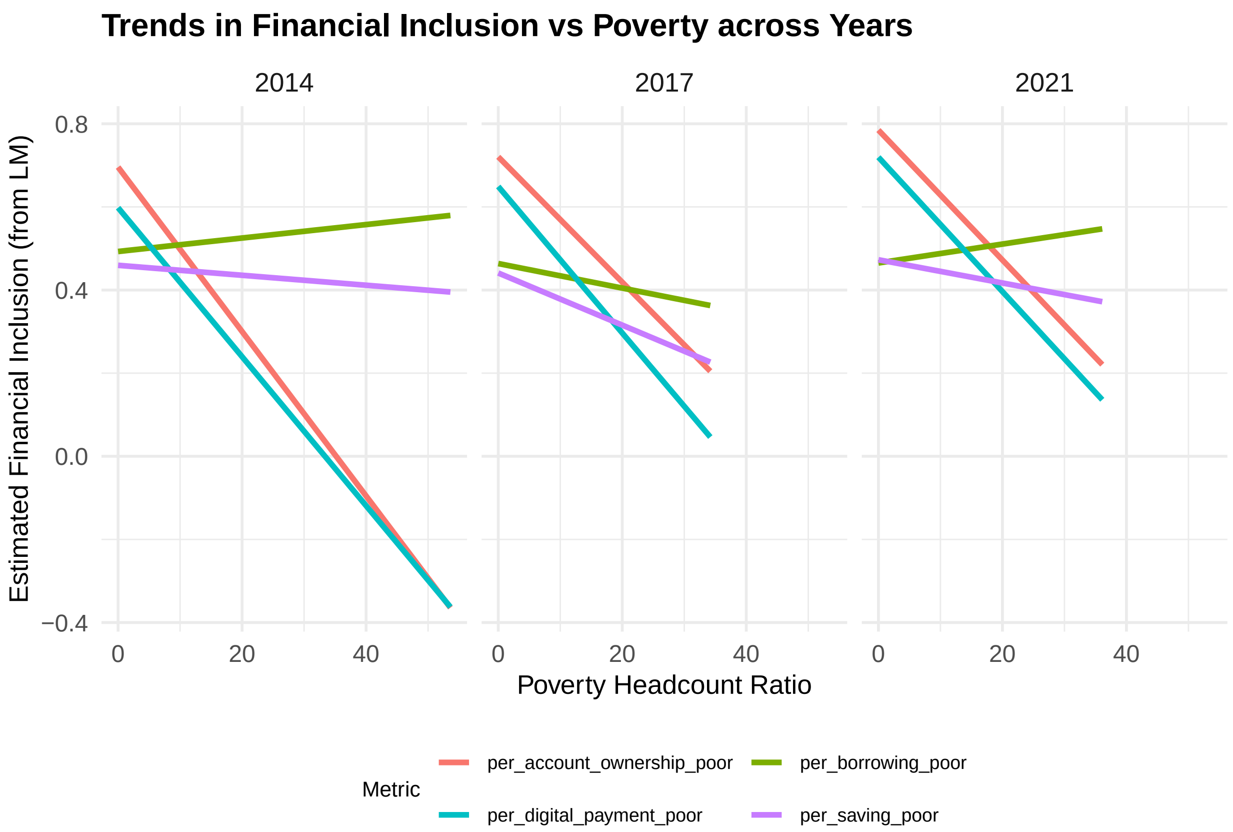
```
## Rows: 796 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): country name, country code, income group, metric
## dbl (7): year, poverty_headcount_ratio, latitude, longitude, gdp, value, avg...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Creating a plot estimating financial inclusion by poverty headcount ratio across years
plot_financial_inclusion_by_poverty <- inclusive_poverty_gdp_long %>%
  ggplot(aes(x = poverty_headcount_ratio, y = value, color = metric)) +
  geom_smooth(method = "lm", se = FALSE, size = 1) +
  facet_wrap(~year) +
  labs(
    title = "Trends in Financial Inclusion vs Poverty across Years",
    x = "Poverty Headcount Ratio",
    y = "Estimated Financial Inclusion (from LM)",
    color = "Metric"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),
    plot.title = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 10),
    legend.position = "bottom",
    legend.title = element_text(size = 8),
    legend.text = element_text(size = 7),
    legend.key.size = unit(0.5, "cm")
  ) +
  guides(
    color = guide_legend(nrow = 2, byrow = TRUE)
  )
```

```
## 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.
```

```
print(plot_financial_inclusion_by_poverty)
```

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



```
ggsave("plot_financial_inclusion_by_poverty.png", plot = plot_financial_inclusion_by_poverty, path = "in
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
# Creating choropleth for financial inclusion in the world
world_map <- st_read("data/ne_10m_admin_0_countries.shp")
```

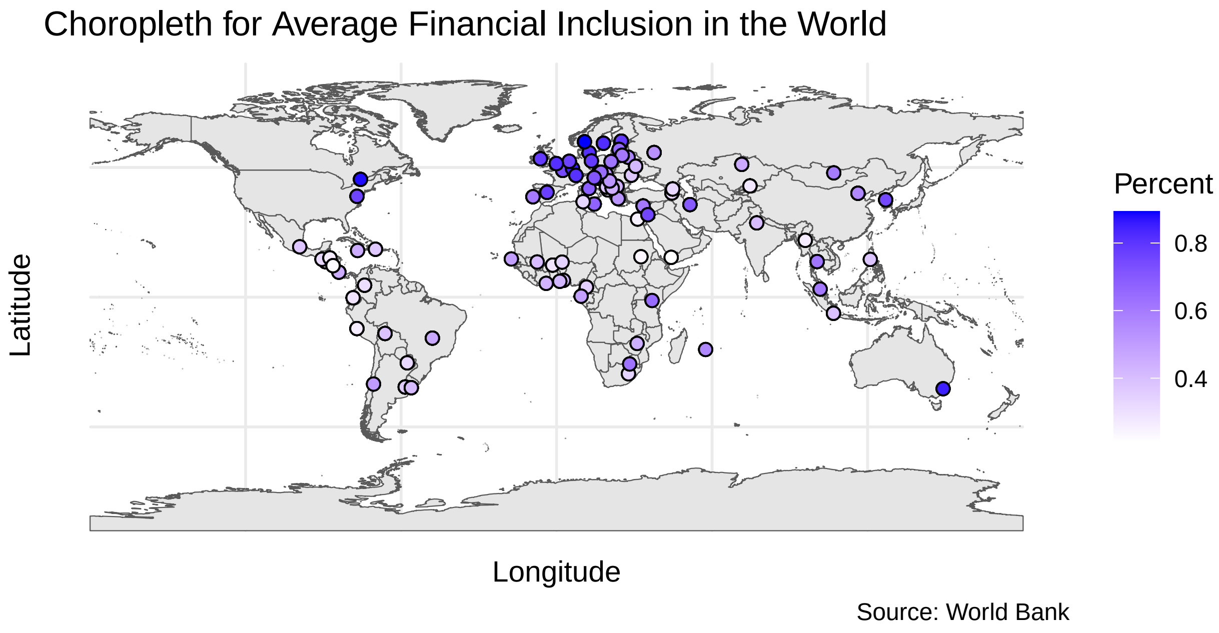
```
## Reading layer `ne_10m_admin_0_countries' from data source
##   `/Users/yeong/Documents/GitHub/R-II_final/data/ne_10m_admin_0_countries.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 258 features and 168 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -90 xmax: 180 ymax: 83.6341
## Geodetic CRS:  WGS 84
```

```
inclusive_poverty_gdp_sf <- st_as_sf(inclusive_poverty_gdp_long,
  coords = c("longitude", "latitude"),
  crs = st_crs(world_map), remove = FALSE
)

choropleth_financial_inclusion <- ggplot() +
  geom_sf(data = world_map) +
```

```r
  geom_sf(data = inclusive_poverty_gdp_sf, aes(fill = avg_value), shape = 21, size = 2) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(
    title = "Choropleth for Average Financial Inclusion in the World",
    fill = "Percent",
    x = "Longitude", y = "Latitude",
    caption = "Source: World Bank"
  ) +
  theme(
    strip.text = element_text(size = 10, face = "bold"),
    plot.title = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 10)
  ) +
  theme_minimal()

print(choropleth_financial_inclusion)
```



```r
ggsave("choropleth_financial_inclusion.png", plot = choropleth_financial_inclusion, path = "images", wi
```

## Shinyapp

```r
# Create scattor plots to select a country and year and visualize how financial inclusion
# for the poor correlates with poverty
```

```r
## Define ui
ui <- fluidPage(
  titlePanel("Financial Inclusion for the Poor vs Poverty"),
  sidebarLayout(
    sidebarPanel(
      tags$img(
        src = "https://d11jve6usk2wa9.cloudfront.net/platform/10747/assets/logo.png",
        height = 80,
        width = 220
      ),
      selectInput(inputId = "country_input", label = "Country", choices = NULL),
      selectInput(inputId = "year_input", label = "Year", choices = NULL)
    ),
    mainPanel(
      plotlyOutput("country_plot")
    )
  )
)

## Define server logic
server <- function(input, output, session) {
  inclusive_poverty_gdp_long <- read_csv("data/inclusive_poverty_gdp_long.csv")

  observe({
    country_choices <- unique(inclusive_poverty_gdp_long$`country name`)
    updateSelectInput(session, "country_input", choices = country_choices)
  })

  ### Reactive filtered data based on user inputted country
  filtered <- reactive({
    data <- inclusive_poverty_gdp_long

    # Filter by country
    if (!is.null(input$country_input) && input$country_input != "") {
      data <- data[data$`country name` == input$country_input, ]
    }

    # Filter by year
    if (!is.null(input$year_input) && input$year_input != "") {
      data <- data[data$year == as.numeric(input$year_input), ]
    }
    print(data)
    data
  })

  ### call the resulting dataframe filtered
  observeEvent(input$country_input, {
    filtered_data <- inclusive_poverty_gdp_long[inclusive_poverty_gdp_long$`country name` == input$coun
    year_choices <- unique(filtered_data$year)
    updateSelectInput(session, "year_input", choices = c("All Years" = "", year_choices))
  })

  ### Render the plot
  output$country_plot <- renderPlotly({
    plot_data <- filtered()

    if (nrow(plot_data) == 0) {
      return(NULL)
    }

    ggplotly(
      ggplot(data = plot_data) +
        geom_point(aes(poverty_headcount_ratio, value, fill = metric)) +
        labs(
          title = paste("Financial Inclusion for", input$country_input, "in", input$year_input),
          x = "Poverty Headcount Ratio",
          y = "Financial Inclusion for income, poorest 40% (% ages 15+)",
          caption = "Source: World Bank"
        ) +
        theme_minimal() +
        theme(
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10)
        )
    )
  })
}

shinyApp(ui = ui, server = server)
```

```
# Text process
## Load text from a PDF and turn into a dataframe
microfinance <- pdf_text("data/Microfinance_poverty_trap.pdf")

parsed <- udpipe(microfinance, "english")

## Define ui
ui_2 <- fluidPage(
  titlePanel("Keyword-specific Sentiment from Microfinance vs Poverty Research"),
  sidebarLayout(
    sidebarPanel(
      tags$img(
        src = "https://d11jve6usk2wa9.cloudfront.net/platform/10747/assets/logo.png",
        height = 80,
        width = 220
      ),
      selectInput(inputId = "keyword_input", label = "Keywords", choices = NULL),
    ),
    mainPanel(
      plotlyOutput("dependency_plot")
    )
  )
)

## Define server
server_2 <- function(input, output, session) {
```

```r
### Extract 10 keywords the most used
keywords <- parsed %>%
  filter(!lemma %in% stop_words$word, !upos %in% c("PUNCT", "CCONJ")) %>%
  anti_join(stop_words, by = c("lemma" = "word")) %>%
  group_by(lemma) %>%
  count(sort = TRUE) %>%
  head(n = 10) %>%
  pull(lemma)

# Populate keyword choices
updateSelectInput(inputId = "keyword_input", choices = keywords)

### Reactive filtered data based on user inputted keyword
filtered <- reactive({
  req(input$keyword_input)
  data <- parsed %>% filter(lemma == input$keyword_input)
  validate(
    need(nrow(data) > 0, "No data found for the selected keyword.")
  )
  data
})

### Reactive keyword-specific sentiment
sentiment <- reactive({
  req(filtered())

  filtered_data <- filtered() %>%
    mutate(
      doc_id = as.character(doc_id),
      page_number = as.numeric(gsub("[^0-9]", "", doc_id))
    ) %>%
    arrange(page_number) %>%
    distinct(doc_id, .keep_all = TRUE)

  sentiment_afinn <- get_sentiments("afinn") %>% rename(afinn = value)

  # Extract dependency of keyword and combine altogether
  children <- filtered_data %>%
    inner_join(
      parsed %>% select(doc_id, head_token_id, lemma),
      by = c("token_id" = "head_token_id", "doc_id" = "doc_id")
    ) %>%
    select(doc_id, page_number, lemma.y) %>%
    rename(word = lemma.y)

  parents <- filtered_data %>%
    inner_join(
      parsed %>% select(doc_id, token_id, lemma),
      by = c("head_token_id" = "token_id", "doc_id" = "doc_id")
    ) %>%
    select(doc_id, page_number, lemma.y) %>%
    rename(word = lemma.y)
    combined <- rbind(children, parents) %>%
      left_join(sentiment_afinn, by = "word") %>%
      group_by(doc_id, page_number) %>%
      summarise(afinn = mean(afinn, na.rm = TRUE)) %>%
      ungroup()

    combined
  })

  output$dependency_plot <- renderPlotly({
    req(sentiment())

    # Adjust page number by starting from 2 to be aligned with the paper's page number
    # and omit the reference pages
    sentiment_data <- sentiment() %>%
      mutate(
        page_number = page_number - min(page_number[-1]) + 1
      ) %>%
      filter(page_number <= 59)

    ggplotly(
      ggplot(sentiment_data, aes(x = page_number, y = afinn)) +
        geom_line(color = "steelblue", size = 1) +
        geom_point(aes(color = afinn), size = 3) +
        scale_x_continuous(
          breaks = seq(1, 59, by = 5),
          labels = seq(1, 59, by = 5)
        ) +
        scale_color_gradient2(
          low = "red", mid = "white", high = "blue", midpoint = 0,
          name = "Sentiment Score"
        ) +
        labs(
          title = paste("Sentiment Trend of", input$keyword_input, "(AFINN)"),
          x = "Page Number",
          y = "Average Sentiment"
        ) +
        theme_minimal() +
        theme(
          axis.text.x = element_text(angle = 45, hjust = 1)
        )
    )
  })
}

## Run the app
shinyApp(ui = ui_2, server = server_2)
```

## Model fitting

```
# Load the prepared dataset
data <- read_csv("data/inclusive_poverty_gdp_model.csv")
```

```
## Rows: 141 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): country name, country code, income group
## dbl (9): year, per_account_ownership_poor, per_saving_poor, per_borrowing_po...
##
## 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.
```

```
# Model fit (Fixed Effects Model)
fe_model <- plm(
  poverty_headcount_ratio ~ per_account_ownership_poor + per_saving_poor +
    per_borrowing_poor + per_digital_payment_poor + gdp,
  data = data,
  index = c("country.name", "year"),
  model = "within"
)

# Cluster standard error
clustered_se <- vcovHC(fe_model, method = "arellano", type = "HC0", cluster = "group")
se_clustered <- sqrt(diag(clustered_se))

# Create a summary table
stargazer(
  fe_model,
  se = list(se_clustered),
  type = "latex",
  out = "model_results.latex",
  dep.var.labels = "Poverty Headcount Ratio",
  covariate.labels = c(
    "Account Ownership (Income, poorest 40% (% ages 15+))",
    "Savings (Income, poorest 40% (% ages 15+))",
    "Borrowing (Income, poorest 40% (% ages 15+))",
    "Digital Payments (Income, poorest 40% (% ages 15+))",
    "Real GDP Growth"
  ),
  title = "Fixed Effects Model with Clustered Standard Errors",
  align = TRUE,
  no.space = TRUE
)
```

```
##
## % Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-
mail: marek.hlavac at gmail.com
## % Date and time: Mon, Jan 06, 2025 - 08:16:22
## % Requires LaTeX packages: dcolumn
## \begin{table}[!htbp] \centering
##   \caption{Fixed Effects Model with Clustered Standard Errors}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lD{.}{.}{-3} }
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{1}{c}{\textit{Dependent variable:}} \\
## \cline{2-2}
## \\[-1.8ex] & \multicolumn{1}{c}{Poverty Headcount Ratio} \\
## \hline \\[-1.8ex]
##  Account Ownership (Income, poorest 40% (% ages 15+)) & 0.505 \\
##   & (3.344) \\
##   Savings (Income, poorest 40% (% ages 15+)) & 0.757 \\
##   & (1.200) \\
##   Borrowing (Income, poorest 40% (% ages 15+)) & 3.524^{*} \\
##   & (1.807) \\
##   Digital Payments (Income, poorest 40% (% ages 15+)) & -3.146 \\
##   & (2.595) \\
##   Real GDP Growth & -0.004 \\
##   & (0.034) \\
##  \hline \\[-1.8ex]
## Observations & \multicolumn{1}{c}{141} \\
## R$^{2}$ & \multicolumn{1}{c}{0.172} \\
## Adjusted R$^{2}$ & \multicolumn{1}{c}{-0.302} \\
## F Statistic & \multicolumn{1}{c}{3.703$^{***}$ (df = 5; 89)} \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{1}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```