

Draft Documents

Team 1 (Hedgehog, Callista, Imtiyaaz, Issac)

2022-11-14

```
# Package names
packages <- c('MASS', 'knitr',
# 'readr', 'tidy whole', 'dplyr', 'ggplot2', 'e1071',
# 'moments', 'corrplot', 'Hmisc',
# 'PerformanceAnalytics', 'mice', 'needs') #
# Install packages not yet installed
# installed_packages <- packages %in%
# rownames(installed.packages()) if
# (any(installed_packages == FALSE)) {
# install.packages(packages[!installed_packages])
# } # Packages loading invisible(lapply(packages,
# library, character.only = TRUE))
# prioritize(dplyr)

library(needs)

needs("dplyr", "MASS", "knitr", "readr", "tidy whole", "ggplot2",
      "e1071", "moments", "corrplot", "Hmisc", "PerformanceAnalytics",
      "mice", "car", "glmmnet")

prioritize(dplyr)
```

1. Introduction (To be done)

2. Data Characteristic

2.1. Nature of Data

The data set is collection The World Bank Data, the variables of interest are extracted from the raw data files and combined into a single data frame for analysis. The final data set includes:

1. **country.code**: Country code
2. **country.name**: Country name
3. **year**: Year
4. **income**: Income class
 - Low income (L)

- Lower middle income (LM)
- Upper middle income (UM)
- High income (H)

5. **reg**: Region

6. **pov**: Poverty headcount ratio based on cut-off value of \$2.15 per day

7. **mpi**: Multidimensional Poverty Index

8. **edu.total**: Total expenditure on education (% of GDP)

9. **edu.pri**: Total expenditure on primary education (% of total education expenditure)

10. **edu.sec**: Total expenditure on secondary education (% of total education expenditure)

11. **edu.ter**: Total expenditure on tertiary education (% of total education expenditure)

12. **hlth**: Total expenditure on health (% of GDP)

13. **mil**: Total expenditure on military (% of GDP)

14. **fdi**: Foreign Direct Investment

15. **lbr.part**: Labour force participation (% of population ages 15+)

16. **unemp**: Unemployment rate

17. **pop.gwth.total**: Total population growth rate

18. **pop.gwth.rural**: Total rural population growth rate

19. **pop.gwth.urban**: Total urban population growth rate

20. **gdp.dflt**: GDP deflator

21. **gdr.eqi**: Gender equality rating

22. **gcf**: Gross Capital Formation

23. **trade**: Trade = import + export (% of GDP)

24. **gdp.pc**: GDP per capita (current US\$)

Data imports and combining:

```
# helper functions
importWDI <- function(filepath, value_name) {
  df <- read_csv(filepath, skip = 4)

  colnames(df) <- tolower(gsub(" ", ".", colnames(df)))
```

```

df <- df %>%
  pivot_longer(5:ncol(.), names_to = "year",
              values_to = "value") %>%
  filter(!is.null(value) & !is.na(value)) %>%
  mutate(country.code = factor(country.code),
         country.name = factor(country.name), year = as.numeric(year)) %>%
  select(country.code, country.name, year, value)

colnames(df)[4] <- value_name

df
}

importRegionClass <- function(filepath) {
  df <- read_csv(filepath, skip = 4)

  colnames(df) <- tolower(gsub(" ", ".", colnames(df)))

  df %>%
    mutate(country.name = factor(country.name),
           region = factor(region)) %>%
    select(country.name, reg = region)
}

importIncomeClass <- function(filepath) {
  df <- read_csv(filepath, skip = 4)

  colnames(df) <- tolower(gsub(" ", ".", colnames(df)))

  df %>%
    pivot_longer(3:ncol(.), names_to = "year",
                values_to = "income") %>%
    filter(!is.null(income) & !is.na(income)) %>%
    mutate(country.code = factor(country.code),
           country.name = factor(country.name), year = as.numeric(year),
           income = factor(income)) %>%
    select(country.code, country.name, year, income)
}

```

```

# import data
setwd("../data")

poverty.headcount <- importWDI("poverty.headcount.215dollar.csv",
                                "pov")
mpi <- importWDI("mpi.csv", "mpi")
education.expenditure.total <- importWDI("total.education.expenditure.csv",
                                           "edu.total")
education.expenditure.primary <- importWDI("primary.education.expenditure.csv",
                                              "edu.pri")
education.expenditure.secondary <- importWDI("secondary.education.expenditure.csv",
                                               "edu.sec")
education.expenditure.tertiary <- importWDI("tertiary.education.expenditure.csv",
                                             "edu.ter")

```

```

health.expenditure <- importWDI("health.expenditure.csv",
                                 "hlth")
military.expenditure <- importWDI("military.expenditure.csv",
                                    "mil")
fdi <- importWDI("fdi.csv", "fdi")
labour.force.participation <- importWDI("labour.force.participation.csv",
                                         "lbr.part")
unemployment.rate <- importWDI("unemployment.csv",
                                  "unemp")
population.growth <- importWDI("population.growth.csv",
                                 "pop.gwth.total")
rural.population.growth <- importWDI("rural.population.growth.csv",
                                       "pop.gwth.rural")
urban.population.growth <- importWDI("urban.population.growth.csv",
                                       "pop.gwth.urban")
gdp.deflator <- importWDI("gdp.deflator.csv", "gdp.dflt")
gender.equality <- importWDI("gender.equality.csv",
                               "gdr.eq1")
gross.capital.formation <- importWDI("gross.capital.formation.csv",
                                         "gcf")
trade <- importWDI("trade.csv", "trade")
region.class <- importRegionClass("region.class.csv")
income.class <- importIncomeClass("income.class.csv")
gdp.pc <- importWDI("gdp.pc.csv", "gdp.pc")

setwd("../src")

```

We found that the data sets collected from [World Bank's Data helpdesk](#) and [The World Bank's Data](#) have different naming convention for certain countries (e.g. “Czechia” vs. “Czechia Republic”). So we need to rename these countries to avoid some error when joining.

Furthermore, WDI’s data sets rate also account for non-country (e.g. country.name = “Low income” or “South Asia”). These special groups are not in our scope of interest, which is national, so we eliminate them.

```

# using poverty.headcount as a naming standard
# (as other data from WDI also use this
# convention) join a subset of data to process
# the names
d <- poverty.headcount %>%
  select(country.name) %>%
  mutate(inPov = T) %>%
  full_join(income.class %>%
              select(country.name) %>%
              mutate(inIncome = T), by = "country.name") %>%
  full_join(region.class %>%
              select(country.name) %>%
              mutate(inReg = T), by = "country.name") %>%
  mutate(inPov = !is.na(inPov), inIncome = !is.na(inIncome),
         inReg = !is.na(inReg))

## # A tibble: 62,759 x 4
##   country.name inPov inIncome inReg

```

```

##      <fct>     <lgl> <lgl>     <lgl>
## 1 Angola      TRUE  TRUE  TRUE
## 2 Angola      TRUE  TRUE  TRUE
## 3 Angola      TRUE  TRUE  TRUE
## 4 Angola      TRUE  TRUE  TRUE
## 5 Angola      TRUE  TRUE  TRUE
## 6 Angola      TRUE  TRUE  TRUE
## 7 Angola      TRUE  TRUE  TRUE
## 8 Angola      TRUE  TRUE  TRUE
## 9 Angola      TRUE  TRUE  TRUE
## 10 Angola     TRUE  TRUE  TRUE
## # ... with 62,749 more rows
## # i Use 'print(n = ...)' to see more rows

```

First, remove special economic groups from `poverty.headcount`. We figured these regions will not appear in `income.class` or `region.class`, so we might find something from looking at the countries **only** appear in `poverty.headcount`.

```

d %>%
  filter(inPov & (!inIncome | !inReg)) %>%
  distinct(country.name)

```

```

## # A tibble: 18 x 1
##   country.name
##      <fct>
## 1 Cote d'Ivoire
## 2 Czechia
## 3 East Asia & Pacific
## 4 Europe & Central Asia
## 5 Fragile and conflict affected situations
## 6 High income
## 7 IDA total
## 8 Latin America & Caribbean
## 9 Low income
## 10 Lower middle income
## 11 Low & middle income
## 12 Middle East & North Africa
## 13 South Asia
## 14 Sub-Saharan Africa
## 15 Sao Tome and Principe
## 16 Turkiye
## 17 Upper middle income
## 18 World

```

Lucky! We can look through these 18 results and compose a list of special regions.

```

spec.reg <- c("Fragile and conflict affected situations",
  "IDA total", "World", "East Asia & Pacific", "Europe & Central Asia",
  "Latin America & Caribbean", "Middle East & North Africa",
  "South Asia", "Sub-Saharan Africa", "Low income",
  "Low & middle income", "Lower middle income", "Upper middle income",
  "High income")

```

Then, we rename those countries with inconsistent naming convention. Since we should only care about countries whose poverty headcount is available, reusing the list generated above, we can identify:

1. Cote d'Ivoire (also Côte d'Ivoire)
2. Czechia (also Czechoslovakia or Czech Republic)
3. Curacao (also Curaçao)
4. Turkiye (formerly known as Turkey, also Türkiye)
5. Sao Tome and Principe (also São Tomé and Príncipe)

```
# mapping standard name and variation
nameMap <- tibble(standard = c("Cote d'Ivoire", "Czechia",
  "Czechia", "Curacao", "Turkiye", "Turkiye", "Sao Tome and Principe"),
  variation = c("Côte d'Ivoire", "Czechoslovakia",
  "Czech Republic", "Curaçao", "Turkey", "Türkiye",
  "São Tomé and Príncipe"))

correctName <- function(name) {
  tibble(name = name) %>%
    left_join(nameMap, by = c(name = "variation")) %>%
    mutate(standard = ifelse(is.na(standard), name,
      standard)) %>%
    select(standard) %>%
    pull()
}

orig.name <- c("Vietnam", "China", "Turkey", "Czechia Republic")
correctName(orig.name)
```

```
## [1] "Vietnam"          "China"           "Turkiye"         "Czechia Republic"
```

Let's test this out!

```
d <- poverty.headcount %>%
  # correct name here
  mutate(country.name = correctName(country.name)) %>%
  select(country.name) %>%
  mutate(inPov = T) %>%
  full_join(income.class %>%
    # correct name here
    mutate(country.name = correctName(country.name)) %>%
    select(country.name) %>%
    mutate(inIncome = T), by = "country.name") %>%
  full_join(region.class %>%
    # correct name here
    mutate(country.name = correctName(country.name)) %>%
    select(country.name) %>%
    mutate(inReg = T), by = "country.name") %>%
  filter(!(country.name %in% spec.reg)) %>%
  mutate(inPov = !is.na(inPov), inIncome = !is.na(inIncome), inReg = !is.na(inReg))

# countries not in region list, but is in Pov list
d %>%
  filter(!inReg & inPov) %>%
```

```

distinct(country.name) %>%
nrow()

## [1] 0

# countries not in income list, but is in Pov list
d %>%
filter(!inIncome & inPov) %>%
distinct(country.name) %>%
nrow()

## [1] 0

```

We are *pretty* confident that there's no inconsistent naming left unprocessed in the data sets.

```

# Rename countries in all data sets.
poverty.headcount <- poverty.headcount %>%
  mutate(country.name = correctName(country.name))
mpi <- mpi %>%
  mutate(country.name = correctName(country.name))
education.expenditure.total <- education.expenditure.total %>%
  mutate(country.name = correctName(country.name))
education.expenditure.primary <- education.expenditure.primary %>%
  mutate(country.name = correctName(country.name))
education.expenditure.secondary <- education.expenditure.secondary %>%
  mutate(country.name = correctName(country.name))
education.expenditure.tertiary <- education.expenditure.tertiary %>%
  mutate(country.name = correctName(country.name))
health.expenditure <- health.expenditure %>%
  mutate(country.name = correctName(country.name))
military.expenditure <- military.expenditure %>%
  mutate(country.name = correctName(country.name))
fdi <- fdi %>%
  mutate(country.name = correctName(country.name))
labour.force.participation <- labour.force.participation %>%
  mutate(country.name = correctName(country.name))
unemployment.rate <- unemployment.rate %>%
  mutate(country.name = correctName(country.name))
population.growth <- population.growth %>%
  mutate(country.name = correctName(country.name))
rural.population.growth <- rural.population.growth %>%
  mutate(country.name = correctName(country.name))
urban.population.growth <- urban.population.growth %>%
  mutate(country.name = correctName(country.name))
gdp.deflator <- gdp.deflator %>%
  mutate(country.name = correctName(country.name))
gender.equality <- gender.equality %>%
  mutate(country.name = correctName(country.name))
gross.capital.formation <- gross.capital.formation %>%
  mutate(country.name = correctName(country.name))
trade <- trade %>%
  mutate(country.name = correctName(country.name))

```

```

region.class <- region.class %>%
  mutate(country.name = correctName(country.name))
income.class <- income.class %>%
  mutate(country.name = correctName(country.name))
gdp.pc <- gdp.pc %>%
  mutate(country.name = correctName(country.name))

```

Join the data

```

countries <- poverty.headcount %>%
  # We used a full join here so we can conduct
  # a separate analysis on mpi later
full_join(mpi, by = c("country.name", "country.code",
  "year")) %>%
  left_join(income.class, c("country.name", "country.code",
    "year")) %>%
  left_join(region.class, by = "country.name") %>%
  left_join(education.expenditure.total, by = c("country.name",
    "country.code", "year")) %>%
  left_join(education.expenditure.primary, by = c("country.name",
    "country.code", "year")) %>%
  left_join(education.expenditure.secondary, by = c("country.name",
    "country.code", "year")) %>%
  left_join(education.expenditure.tertiary, by = c("country.name",
    "country.code", "year")) %>%
  left_join(health.expenditure, by = c("country.name",
    "country.code", "year")) %>%
  left_join(military.expenditure, by = c("country.name",
    "country.code", "year")) %>%
  left_join(fdi, by = c("country.name", "country.code",
    "year")) %>%
  left_join/labour.force.participation, by = c("country.name",
    "country.code", "year")) %>%
  left_join(unemployment.rate, by = c("country.name",
    "country.code", "year")) %>%
  left_join/population.growth, by = c("country.name",
    "country.code", "year")) %>%
  left_join(rural.population.growth, by = c("country.name",
    "country.code", "year")) %>%
  left_join(urban.population.growth, by = c("country.name",
    "country.code", "year")) %>%
  left_join(gdp.deflator, by = c("country.name",
    "country.code", "year")) %>%
  left_join(gender.equality, by = c("country.name",
    "country.code", "year")) %>%
  left_join(gross.capital.formation, by = c("country.name",
    "country.code", "year")) %>%
  left_join(trade, by = c("country.name", "country.code",
    "year")) %>%
  left_join(gdp.pc, by = c("country.name", "country.code",
    "year")) %>%
  # filter special groups
filter(!(country.name %in% spec.reg))

```

Data preview

```
head(countries)
```

```
## # A tibble: 6 x 24
##   count~1 count~2 year   pov   mpi income reg   edu.t~3 edu.pri edu.sec edu.ter
##   <fct>  <chr>   <dbl> <dbl> <dbl> <fct>  <fct>  <dbl> <dbl> <dbl> <dbl>
## 1 AGO    Angola  2000  21.4  NA L    Sub--  2.61   NA    NA    NA
## 2 AGO    Angola  2008  14.6  NA LM   Sub--  2.69   NA    NA    NA
## 3 AGO    Angola  2018  31.1  NA LM   Sub--  2.04   NA    NA    NA
## 4 ALB    Albania 1996   0.5  NA LM   Euro-- 3.08   NA    NA    NA
## 5 ALB    Albania 2002   1.1  NA LM   Euro-- 3.12   NA    NA    NA
## 6 ALB    Albania 2005   0.6  NA LM   Euro-- 3.28   NA    NA    NA
## # ... with 13 more variables: hlth <dbl>, mil <dbl>, fdi <dbl>, lbr.part <dbl>,
## #   unemp <dbl>, pop.gwth.total <dbl>, pop.gwth.rural <dbl>,
## #   pop.gwth.urban <dbl>, gdp.dflt <dbl>, gdr.eql <dbl>, gcf <dbl>,
## #   trade <dbl>, gdp.pc <dbl>, and abbreviated variable names 1: country.code,
## #   2: country.name, 3: edu.total
## # i Use 'colnames()' to see all variable names
```

```
str(countries)
```

```
### tibble [1,901 x 24] (S3: tbl_df/tbl/data.frame)
### $ country.code : Factor w/ 272 levels "AGO","ALB","ARE",...: 1 1 1 2 2 2 2 2 2 ...
### $ country.name : chr [1:1901] "Angola" "Angola" "Angola" "Albania" ...
### $ year         : num [1:1901] 2000 2008 2018 1996 2002 ...
### $ pov          : num [1:1901] 21.4 14.6 31.1 0.5 1.1 0.6 0.2 0.6 1 0.1 ...
### $ mpi          : num [1:1901] NA NA NA NA NA NA NA NA NA ...
### $ income        : Factor w/ 4 levels "H","L","LM","UM": 2 3 3 3 3 3 3 4 4 4 ...
### $ reg           : Factor w/ 7 levels "East Asia & Pacific",...: 7 7 7 2 2 2 2 2 2 ...
### $ edu.total     : num [1:1901] 2.61 2.69 2.04 3.08 3.12 ...
### $ edu.pri       : num [1:1901] NA NA NA NA NA ...
### $ edu.sec       : num [1:1901] NA NA NA NA NA ...
### $ edu.ter       : num [1:1901] NA NA NA NA NA ...
### $ hlth          : num [1:1901] 1.91 3.32 2.54 NA 6.91 ...
### $ mil           : num [1:1901] 6.39 3.57 1.87 1.38 1.32 ...
### $ fdi           : num [1:1901] 8.79e+08 1.68e+09 -6.46e+09 9.01e+07 1.35e+08 ...
### $ lbr.part      : num [1:1901] NA NA NA 38.8 59.6 ...
### $ unemp         : num [1:1901] NA NA NA 12.3 15.8 ...
### $ pop.gwth.total: num [1:1901] 3.277 3.711 3.276 -0.622 -0.3 ...
### $ pop.gwth.rural: num [1:1901] 0.921 1.91 1.338 -1.546 -2.169 ...
### $ pop.gwth.urban: num [1:1901] 5.682 5.02 4.312 0.812 2.181 ...
### $ gdp.dflt      : num [1:1901] 418.02 19.37 28.17 38.17 3.65 ...
### $ gdr.eql       : num [1:1901] NA 3 NA NA NA 4 NA NA NA NA ...
### $ gcf           : num [1:1901] 30.5 30.8 17.9 18.1 35.3 ...
### $ trade          : num [1:1901] 152.5 121.4 66.4 44.9 68.5 ...
### $ gdp.pc         : num [1:1901] 557 4081 2525 1010 1425 ...
```

```
summary(countries)
```

| | country.code | country.name | year | pov |
|--------|--------------|--------------|------------|-------------|
| ## BRA | : 36 | Length:1901 | Min. :1967 | Min. : 0.00 |

```

##   CRI    : 34   Class :character   1st Qu.:2002   1st Qu.: 0.20
##   ARG    : 32   Mode  :character   Median :2009   Median : 1.50
##   USA    : 32                           Mean   :2007   Mean   :10.04
##   DEU    : 30                           3rd Qu.:2014   3rd Qu.:11.60
##   HND    : 30                           Max.   :2021   Max.   :91.50
##   (Other):1707                         NA's   :58
##   mpi      income
##   Min.   : 2.37   H   :644   East Asia & Pacific   :167   Min.   : 1.033
##   1st Qu.:18.30   L   :253   Europe & Central Asia   :883   1st Qu.: 3.522
##   Median :24.80   LM  :501   Latin America & Caribbean :416   Median : 4.519
##   Mean   :27.06   UM  :438   Middle East & North Africa:107   Mean   : 4.582
##   3rd Qu.:33.30   NA's: 65   North America           : 50   3rd Qu.: 5.457
##   Max.   :74.20                           South Asia           : 61   Max.   :15.750
##   NA's   :1446                           Sub-Saharan Africa   :217   NA's   :596
##   edu.pri    edu.sec    edu.ter      hlth
##   Min.   : 0.6578   Min.   : 2.724   Min.   : 0.00   Min.   : 1.718
##   1st Qu.:24.0269   1st Qu.:30.138   1st Qu.:16.61   1st Qu.: 5.151
##   Median :30.4730   Median :35.713   Median :20.59   Median : 6.914
##   Mean   :31.6633   Mean   :35.630   Mean   :20.96   Mean   : 6.975
##   3rd Qu.:38.3324   3rd Qu.:41.380   3rd Qu.:25.14   3rd Qu.: 8.565
##   Max.   :70.0950   Max.   :71.587   Max.   :50.44   Max.   :17.733
##   NA's   :1090     NA's   :1094     NA's   :963     NA's   :464
##   mil      fdi      lbr.part    unemp
##   Min.   : 0.000   Min.   :-3.444e+11   Min.   :30.50   Min.   : 0.250
##   1st Qu.: 1.042   1st Qu.: 2.979e+08   1st Qu.:56.17   1st Qu.: 4.513
##   Median : 1.468   Median : 1.709e+09   Median :61.18   Median : 6.880
##   Mean   : 1.787   Mean   : 1.598e+10   Mean   :60.78   Mean   : 8.145
##   3rd Qu.: 2.103   3rd Qu.: 9.821e+09   3rd Qu.:65.49   3rd Qu.:10.078
##   Max.   :19.385   Max.   : 7.338e+11   Max.   :93.00   Max.   :49.700
##   NA's   :105     NA's   :17       NA's   :320     NA's   :267
##   pop.gwth.total pop.gwth.rural pop.gwth.urban gdp.dflt
##   Min.   :-3.6295   Min.   :-8.56066   Min.   :-4.078   Min.   : -26.300
##   1st Qu.: 0.2656   1st Qu.: -0.85664   1st Qu.: 0.510   1st Qu.: 1.696
##   Median : 1.0318   Median : -0.02461   Median : 1.484   Median : 3.865
##   Mean   : 1.0565   Mean   : 0.00083   Mean   : 1.691   Mean   : 15.831
##   3rd Qu.: 1.7761   3rd Qu.: 0.96362   3rd Qu.: 2.657   3rd Qu.: 8.537
##   Max.   : 5.6145   Max.   : 4.59686   Max.   :13.805   Max.   :3333.585
##   NA's   :14       NA's   :13       NA's   :18
##   gdr.eql    gcf      trade      gdp.pc
##   Min.   :1.500   Min.   : 0.00   Min.   : 1.378   Min.   : 119.7
##   1st Qu.:3.000   1st Qu.:19.65   1st Qu.: 51.063   1st Qu.: 1927.9
##   Median :3.500   Median :22.73   Median : 73.496   Median : 6032.1
##   Mean   :3.592   Mean   :23.88   Mean   : 84.429   Mean   : 15219.5
##   3rd Qu.:4.000   3rd Qu.:26.76   3rd Qu.:105.462   3rd Qu.: 21490.4
##   Max.   :5.000   Max.   :69.48   Max.   :380.104   Max.   :123678.7
##   NA's   :1635    NA's   :72       NA's   :57       NA's   :8

```

There's no NA in `reg`, which is a sign that all naming in the data is remedied. There's some expected NAs in `income` and `pov`, as these data are collected by year. There's a substantial amount of missing data in `mpi`, as this is a relative new concept. We will address the nature, and processing of missing data in the next sections.

2.2. Missing values

As observed from the summary above, the data set contains a lot of missing values in some of the variables.

```
mean(is.na(countries))
```

```
## [1] 0.1819656
```

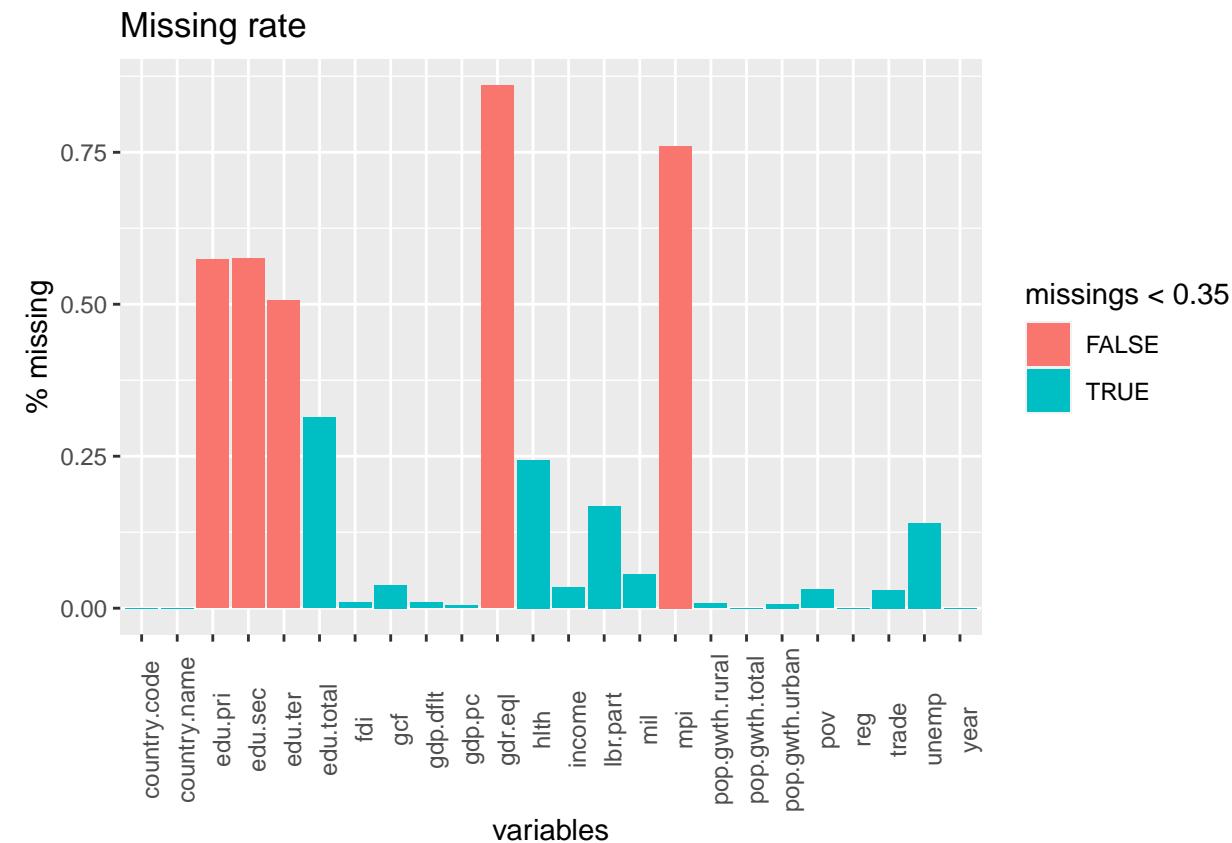
About 19% of the data set is missing.

```
nCompleteObs <- sum(complete.cases(countries))
print(paste("No. of complete cases:", nCompleteObs))
```

```
## [1] "No. of complete cases: 3"
```

There are only 3 complete cases where all the variable is available. This is nowhere near acceptable to conduct any meaningful analysis. Therefore, we need to eliminate some variables for a more balance data set.

```
missings <- colMeans(is.na(countries))
ggplot(mapping = aes(x = names(missings), y = missings,
  fill = missings < 0.35)) + geom_bar(stat = "identity") +
  ggtitle("Missing rate") + xlab("variables") + ylab("% missing") +
  theme(axis.text.x = element_text(size = 9, angle = 90))
```



```
missings[missings > 0.35]
```

```
##      mpi   edu.pri   edu.sec   edu.ter   gdr.eql  
## 0.7606523 0.5733824 0.5754866 0.5065755 0.8600736
```

There are 5 variables with missing rate >35%.

expenditure in primary, secondary, and tertiary education can be very useful and relevant information to predict poverty reduction (Akbar et al. 2019). However, we would like to exclude these variables from some first analyses to make use of the richer set of data. We can conduct a separate analysis with these variable to gain more insight.

```
# variables with high missing rate  
hMiss <- names(missings[missings > 0.35])  
# exclude these variables in countries1  
countries1 <- countries %>%  
  select(!hMiss) %>%  
  filter(!is.na(pov))
```

```
## Note: Using an external vector in selections is ambiguous.  
## i Use 'all_of(hMiss)' instead of 'hMiss' to silence this message.  
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This message is displayed once per session.
```

```
str(countries1)
```

```
## tibble [1,843 x 19] (S3:tbl_df/tbl/data.frame)  
## $ country.code : Factor w/ 272 levels "AGO", "ALB", "ARE", ... : 1 1 1 2 2 2 2 2 2 2 ...  
## $ country.name : chr [1:1843] "Angola" "Angola" "Angola" "Albania" ...  
## $ year : num [1:1843] 2000 2008 2018 1996 2002 ...  
## $ pov : num [1:1843] 21.4 14.6 31.1 0.5 1.1 0.6 0.2 0.6 1 0.1 ...  
## $ income : Factor w/ 4 levels "H", "L", "LM", "UM": 2 3 3 3 3 3 3 4 4 4 ...  
## $ reg : Factor w/ 7 levels "East Asia & Pacific", ... : 7 7 7 2 2 2 2 2 2 2 ...  
## $ edu.total : num [1:1843] 2.61 2.69 2.04 3.08 3.12 ...  
## $ hlth : num [1:1843] 1.91 3.32 2.54 NA 6.91 ...  
## $ mil : num [1:1843] 6.39 3.57 1.87 1.38 1.32 ...  
## $ fdi : num [1:1843] 8.79e+08 1.68e+09 -6.46e+09 9.01e+07 1.35e+08 ...  
## $ lbr.part : num [1:1843] NA NA NA 38.8 59.6 ...  
## $ unemp : num [1:1843] NA NA NA 12.3 15.8 ...  
## $ pop.gwth.total: num [1:1843] 3.277 3.711 3.276 -0.622 -0.3 ...  
## $ pop.gwth.rural: num [1:1843] 0.921 1.91 1.338 -1.546 -2.169 ...  
## $ pop.gwth.urban: num [1:1843] 5.682 5.02 4.312 0.812 2.181 ...  
## $ gdp.dflt : num [1:1843] 418.02 19.37 28.17 38.17 3.65 ...  
## $ gcf : num [1:1843] 30.5 30.8 17.9 18.1 35.3 ...  
## $ trade : num [1:1843] 152.5 121.4 66.4 44.9 68.5 ...  
## $ gdp.pc : num [1:1843] 557 4081 2525 1010 1425 ...
```

Re-evaluate the `countries1` set.

```
mean(is.na(countries1))
```

```
## [1] 0.05471628
```

```

sum(complete.cases(countries1))

## [1] 937

mean(complete.cases(countries1))

## [1] 0.5084102

```

On average, each column has 6% missing rate, results in 937 complete data point (i.e. 49%). This can be a sufficient number for the analysis. However, the missing data can induce loss of power due to the reduced sample size, and some other biases depending on which variables is missing.

```

# complete rate of data by regions
countries1 %>%
  mutate(isComplete = complete.cases(.)) %>%
  group_by(reg) %>%
  summarise(complete.rate = mean(isComplete)) %>%
  arrange(desc(complete.rate))

## # A tibble: 7 x 2
##   reg           complete.rate
##   <fct>          <dbl>
## 1 Europe & Central Asia    0.646
## 2 Latin America & Caribbean 0.505
## 3 Middle East & North Africa 0.462
## 4 East Asia & Pacific      0.437
## 5 South Asia             0.245
## 6 Sub-Saharan Africa       0.192
## 7 North America            0.14

```

Countries from North America, Sub-Saharan Africa, and South Asia have the highest rate of missing data. We suspect that Sub-Saharan Africa, and South Asia are comparably less accessible regions. We also know that Americans don't like filling out forms, so their high rate of missing data is understandable as well.

Still, we need to find a way to address this issue. we propose several approaches:

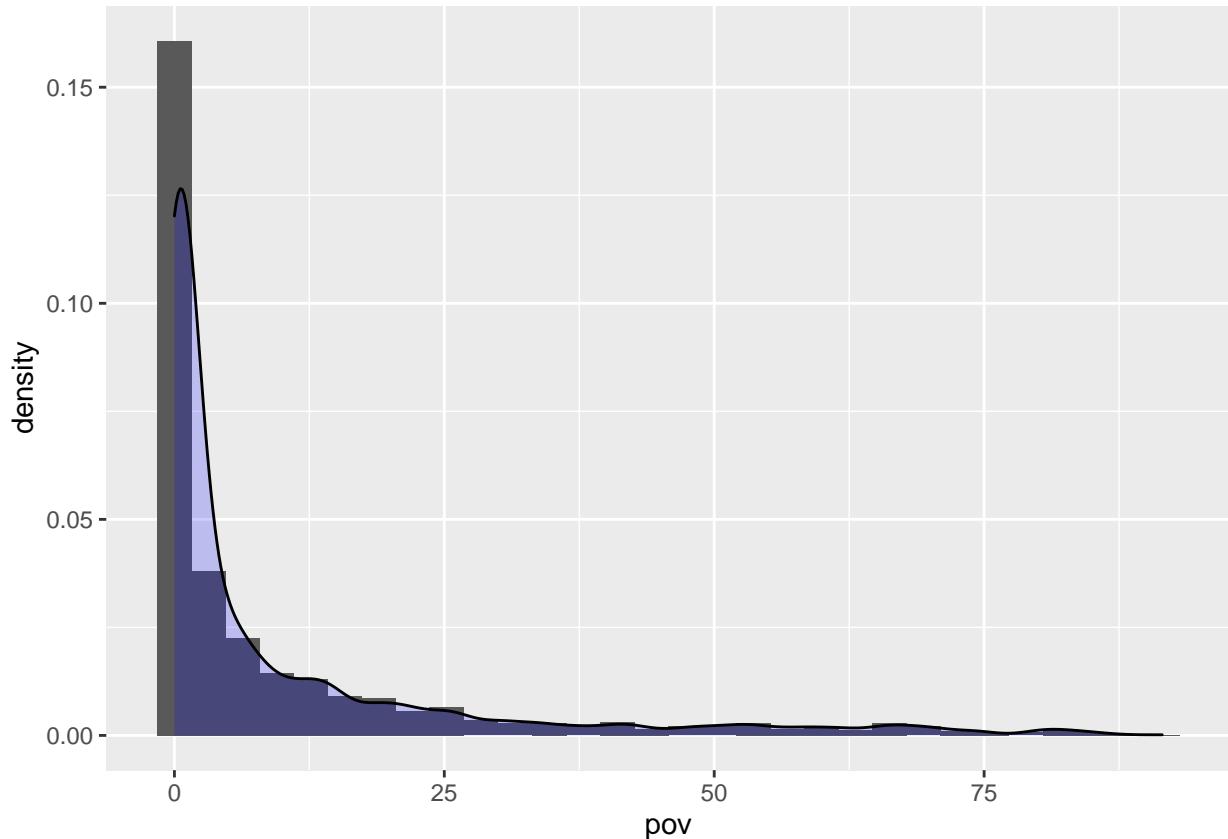
1. **Use complete cases:** Only use the complete cases for the analysis. This is a straightforward approach, but doesn't resolve the bias resulted from the mass loss of data.
2. **Selectively remove variables with high missing rate:** The same as we did before, but this process should be carried out carefully as we run the chance of dropping an important variable.
3. **Update the data set as we select variables:** As we drop insignificant variables (in backward selection), the number of NAs are changed as well. We can utilize the extra complete cases to build the next model in the steps.
4. **Imputation:** The idea is to replace the missing observations on the response or the predictors with artificial values that try to preserve the data set structure. This is a quite complex topic of its own, but we think why not. You can read more from Arel-Bundock and Pelc (2018).

2.3. Descriptive Analytics

Distribution of the predicted variable pov

```
ggplot(countries, aes(x = pov)) + geom_histogram(aes(y = ..density..)) +  
  geom_density(alpha = 0.2, fill = "blue")
```

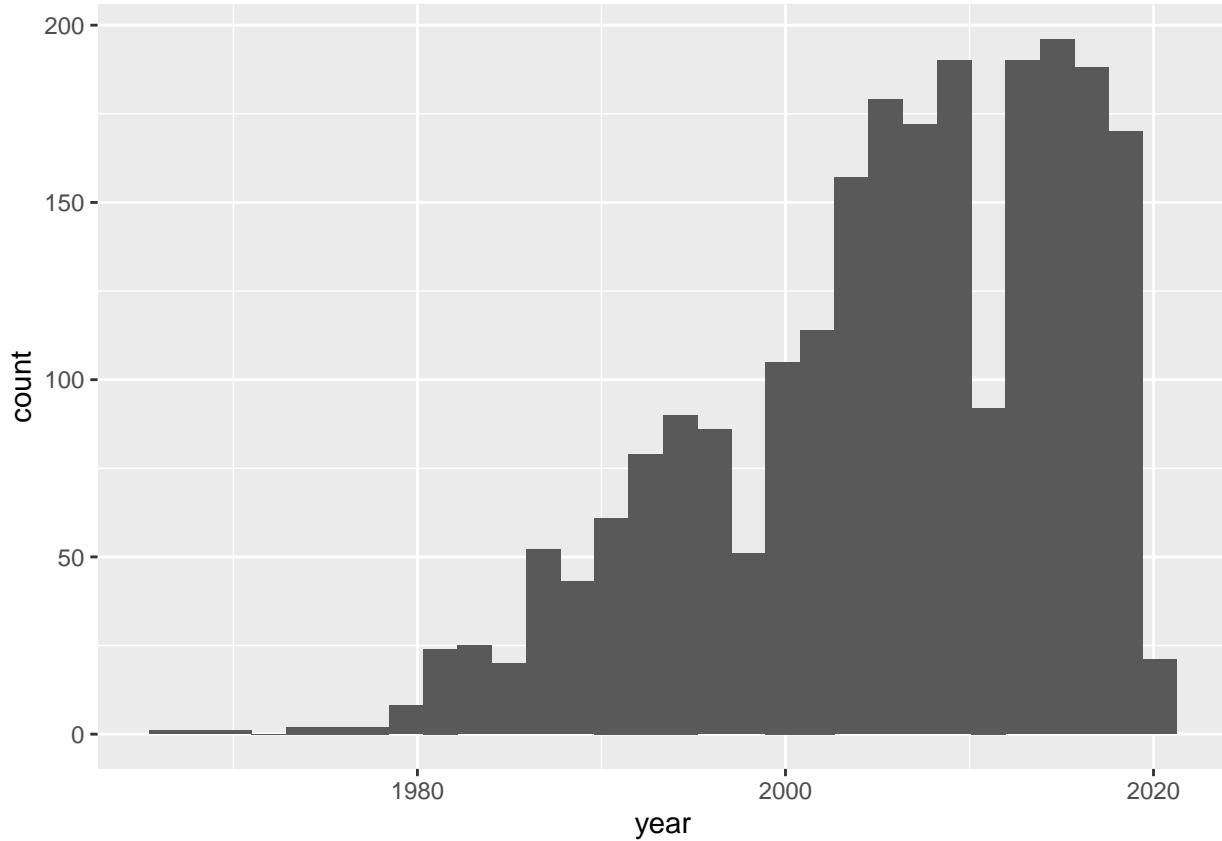
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The graph displays a decreasing rate as poverty indicator increasing. This might not be representative of the current state of poverty in the world, but of the number presented in our data. For example, more recent data is likely to be more inclusive than ancient data, when poverty is more prevalent. We should look at data from the same period.

```
# number of data point available from 1967 to  
# 2021  
ggplot(poverty.headcount, aes(x = year)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

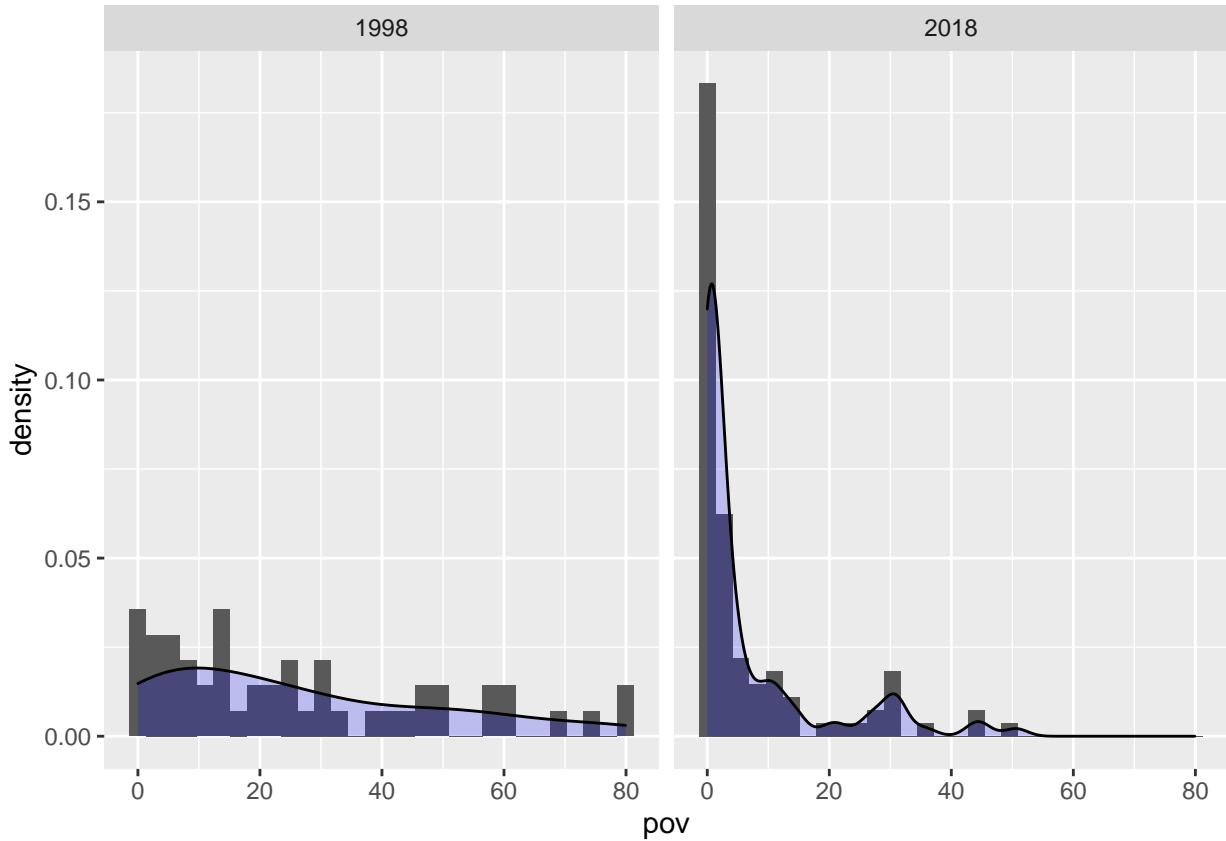


```
# pov data from 1998 and 2018
pov.98.18 <- poverty.headcount %>%
  filter(year == 1998 | year == 2018)
pov.98.18 %>%
  group_by(year) %>%
  summarise(sum = n())

## # A tibble: 2 x 2
##   year     sum
##   <dbl> <int>
## 1 1998     51
## 2 2018     99

ggplot(pov.98.18, aes(x = pov)) + geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.2, fill = "blue") + facet_grid(cols = vars(year))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The graph for 1998 has a much gentler slope, meaning poverty was more popular during that time, as predicted from our intuition. What about the general progress of the world?

```
# Re-import pov and only take special regions
# geographic
geo.regs <- c("EAS", "ECS", "LCN", "MEA", "SAS", "SSF",
             "WLD")
# economics
eco.regs <- c("HIC", "LIC", "LMC", "LMY", "UMC")

pov.reg <- importWDI("../data/poverty.headcount.215dollar.csv",
                      "pov") %>%
  filter(country.code %in% c(geo.regs, eco.regs)) %>%
  mutate(type = ifelse(country.code %in% geo.regs,
                       "Geographic", "Economics"))

## New names:
## Rows: 266 Columns: 67
## -- Column specification
## ----- Delimiter: ","
## (4): Country Name, Country Code, Indicator Name, Indicator Code dbl (50): 1967,
## 1969, 1971, 1974, 1975, 1977, 1978, 1979, 1980, 1981, 1982, ... lgl (13): 1960,
## 1961, 1962, 1963, 1964, 1965, 1966, 1968, 1970, 1972, 1973, ...
## 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.
## * ' -> '...67'
```

```

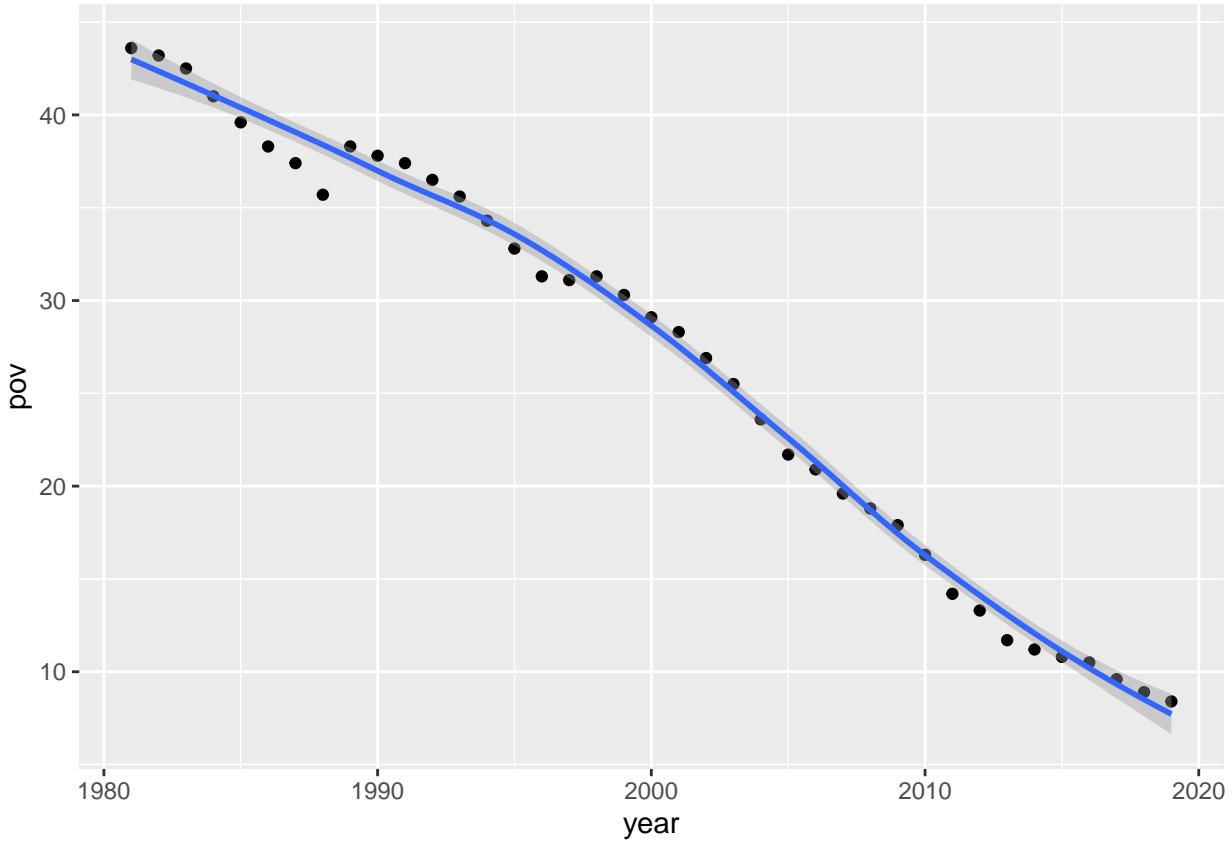
pov.reg %>%
  distinct(country.code, country.name, type) %>%
  arrange(type)

## # A tibble: 12 x 3
##   country.code country.name      type
##   <fct>        <fct>          <chr>
## 1 HIC          High income    Economics
## 2 LIC          Low income     Economics
## 3 LMC          Lower middle income Economics
## 4 LMY          Low & middle income Economics
## 5 UMC          Upper middle income Economics
## 6 EAS          East Asia & Pacific Geographic
## 7 ECS          Europe & Central Asia Geographic
## 8 LCN          Latin America & Caribbean Geographic
## 9 MEA          Middle East & North Africa Geographic
## 10 SAS         South Asia      Geographic
## 11 SSF         Sub-Saharan Africa Geographic
## 12 WLD         World          Geographic

# World
ggplot(pov.reg %>%
  filter(country.code == "WLD"), aes(x = year, y = pov)) +
  geom_point() + geom_smooth()

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

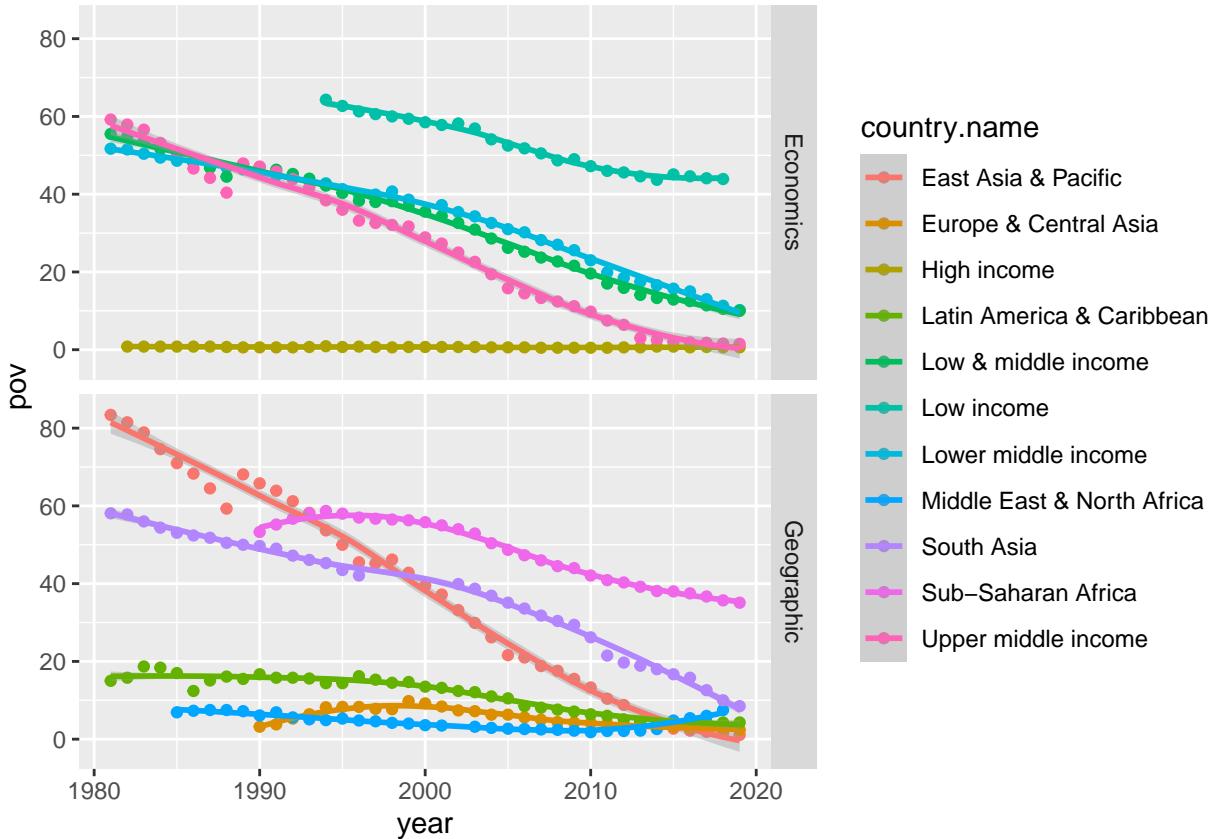
```



An overall very steady decrease of poverty. How about each region?

```
ggplot(pov.reg %>%
  filter(country.code != "WLD"), aes(x = year, y = pov,
  color = country.name)) + geom_point() + geom_smooth() +
  facet_grid(rows = vars(type))

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



There's a general steady, but distinct decline of poverty over time in each type region of respective type. Latin America & Caribbean, Europe & Central Asia, Middle East & North Africa, and High Income group has a more gradual decline as they are not very poor to begin with.

Among the income groups, Low & Middle Income, Lower & Middle Income, and Upper Middle Income have quite similar in term of poverty indicator and slope over the year. While these values vary greatly among different geographical regions.

Let's see some important statistics

```
stats <- poverty.headcount %>%
  summarise(count = n(), skewness = skewness(pov),
           kurtosis = kurtosis(pov), std.deviation = sd(pov))

kable(stats)
```

| | count | skewness | kurtosis | std.deviation |
|--|-------|----------|----------|---------------|
| | 2322 | 1.598182 | 1.661128 | 19.48582 |

2.4. Data Source

- `poverty.headcount`
- `mpi`
- `education.expenditure.primary`
- `education.expenditure.secondary`

- education.expenditure.tertiary
- education.expenditure.total
- health.expenditure
- military.expenditure
- fdi
- unemployment.rate
- labour.force.participation
- gender.equality
- population.growth
- urban.population.growth
- rural.population.growth
- gdp.deflator
- gross.capital.formation
- trade
- region.class
- income.class
- gross.capital.formation

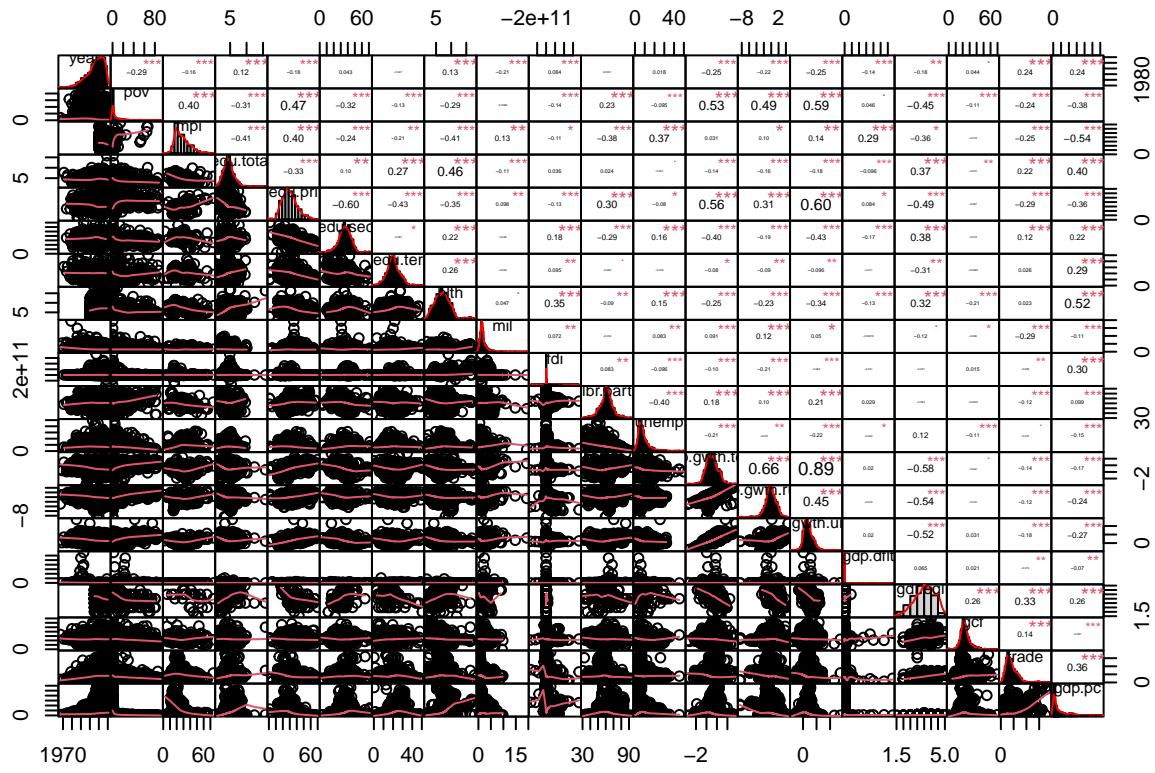
3. Model Selection and Interpretation

3.1. Assumption Check

We can conduct some preliminary checks on linearity and correlation of predictors to have a better picture of the data. **Checklist 1. Linear relationship:**

Use correlation matrix to check linearity

```
# select only numeric data
countries.num <- countries %>%
  select(where(is.numeric))
chart.Correlation(countries.num, histogram = TRUE,
  pch = 19)
```



Which is utterly intelligible, but a majority of the fitted lines are linear at first glance. We should render separate graphs for the relationship between pov & other variables.

```
# lengthen data table to variable-value pairs,
# with pov as predicted variable and others as
# predictive
countries.num2 <- countries.num %>%
  pivot_longer(cols = 3:ncol(.), names_to = "variable",
              values_to = "value") %>%
  filter(!is.na(value) & !is.na(pov))

drawGraph <- function(indvar, data) {
  ggplot(data %>%
    filter(variable == indvar), aes(x = value,
                                     y = pov)) + geom_point() + geom_smooth(method = lm) +
    labs(title = paste("Relationship pov ~", indvar),
         x = indvar, y = "pov")
}

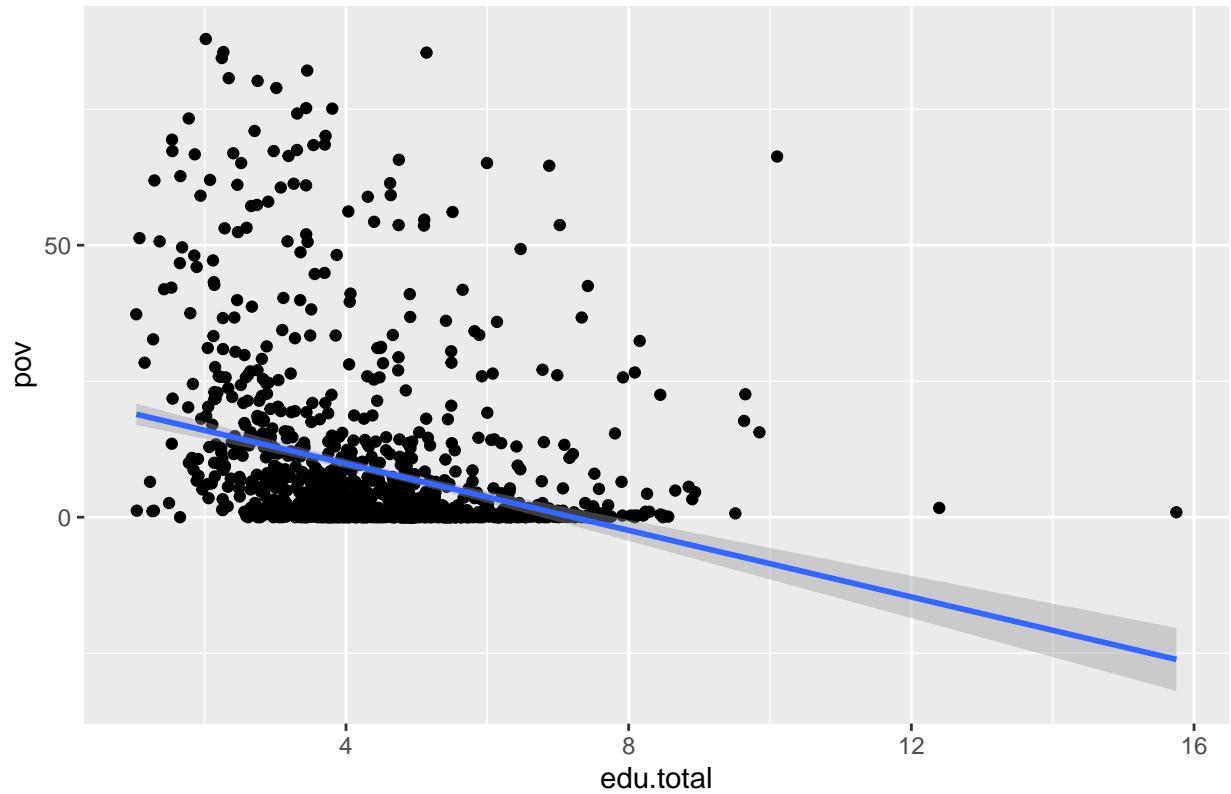
ivs <- distinct(countries.num2, variable)[[1]]

for (indvar in ivs) {
  print(ggplot(countries.num2 %>%
    filter(variable == indvar), aes(x = value,
                                     y = pov)) + geom_point() + geom_smooth(method = lm) +
    labs(title = paste("Relationship pov ~", indvar),
         x = indvar, y = "pov"))
```

```
}
```

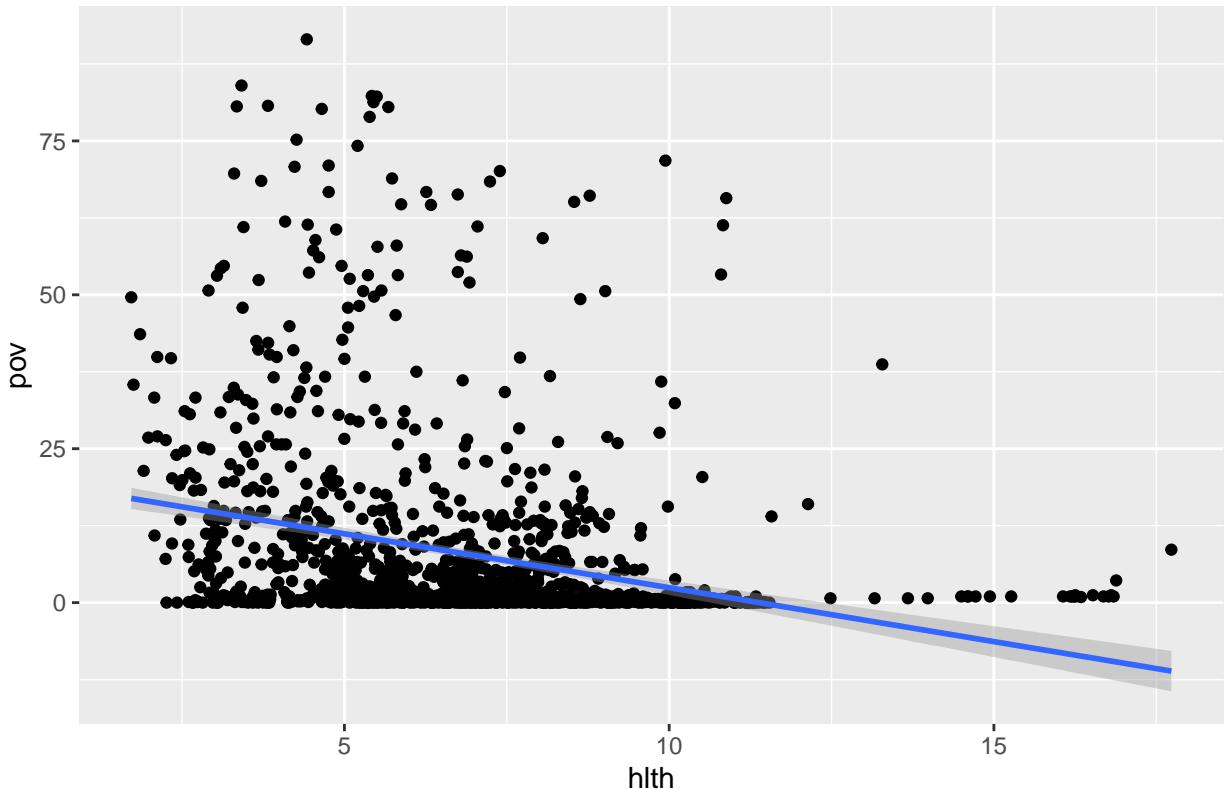
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{edu.total}$



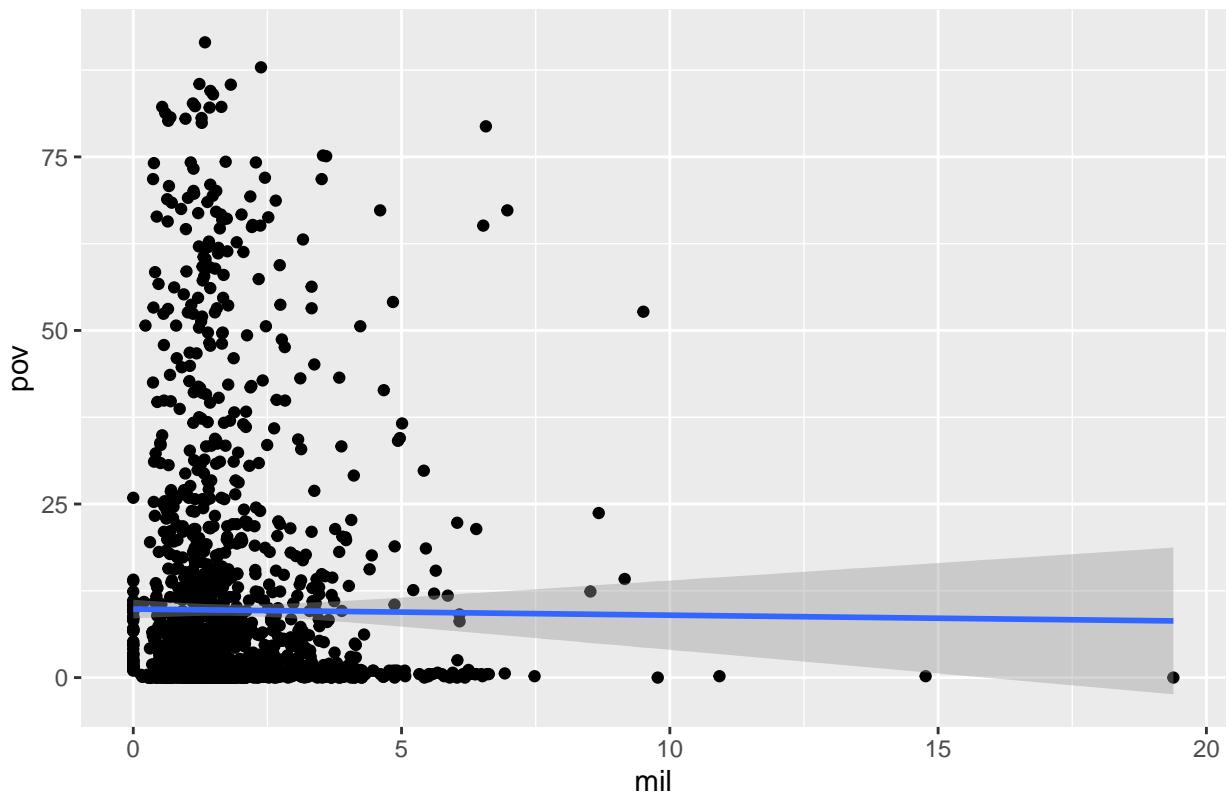
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{hlth}$



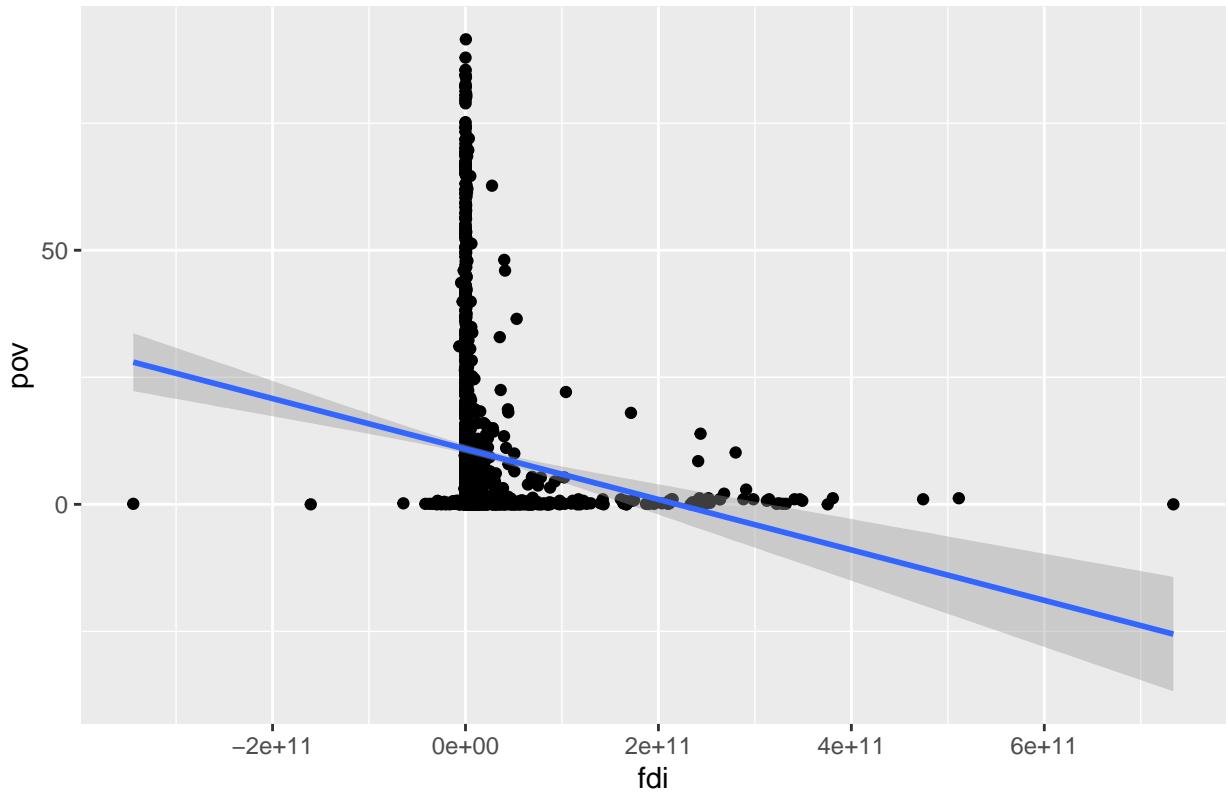
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{mil}$



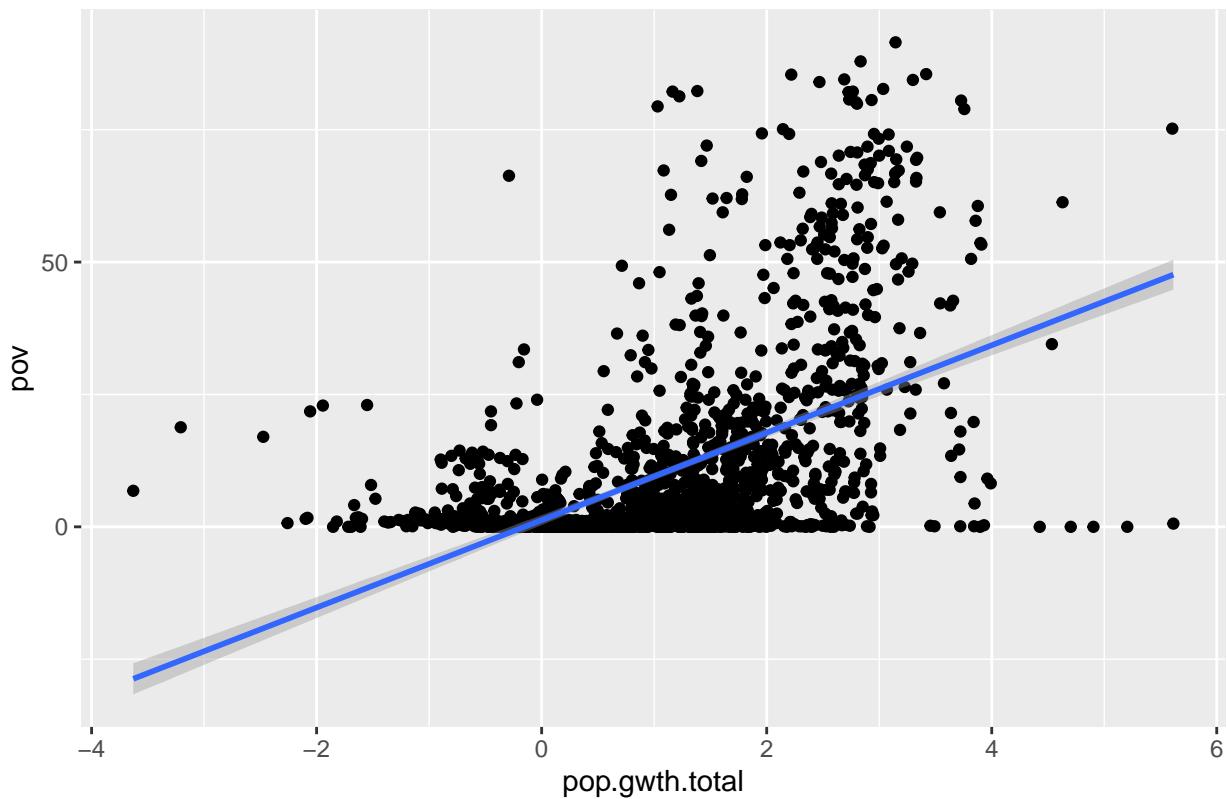
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship pov ~ fdi



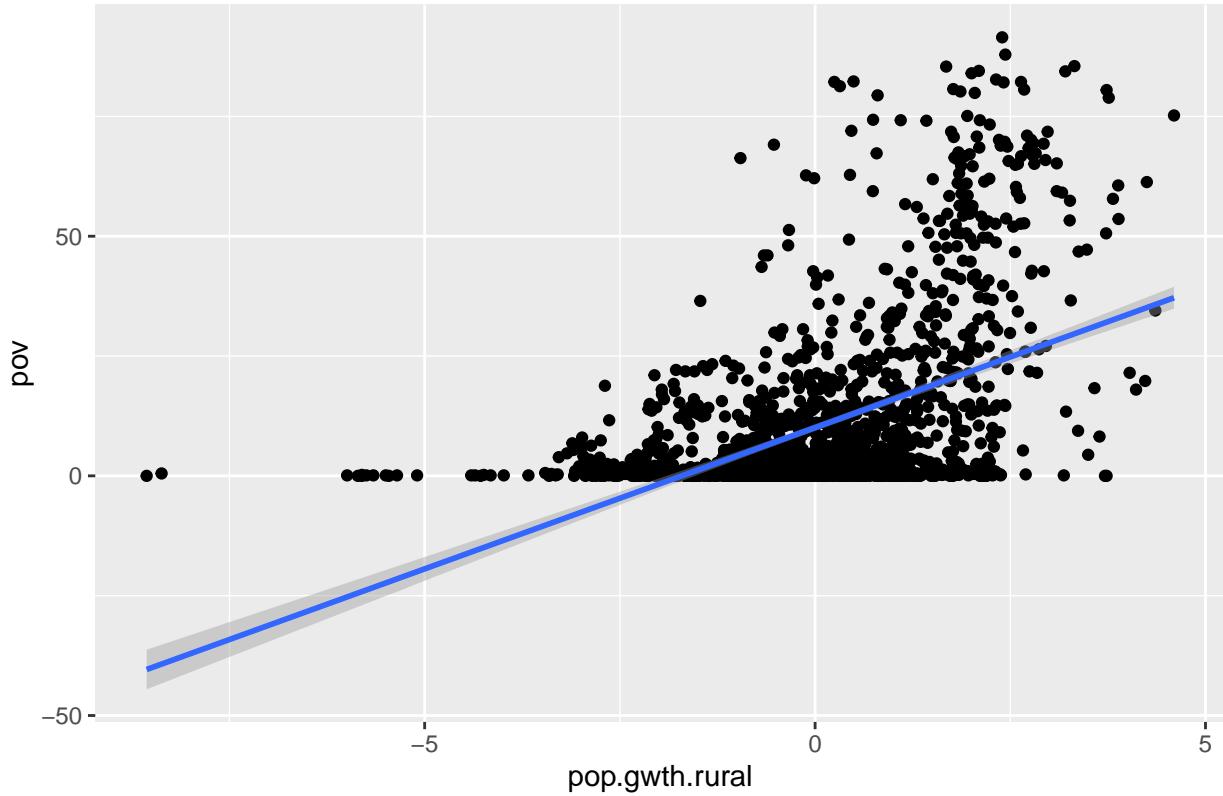
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{pop.gwth.total}$



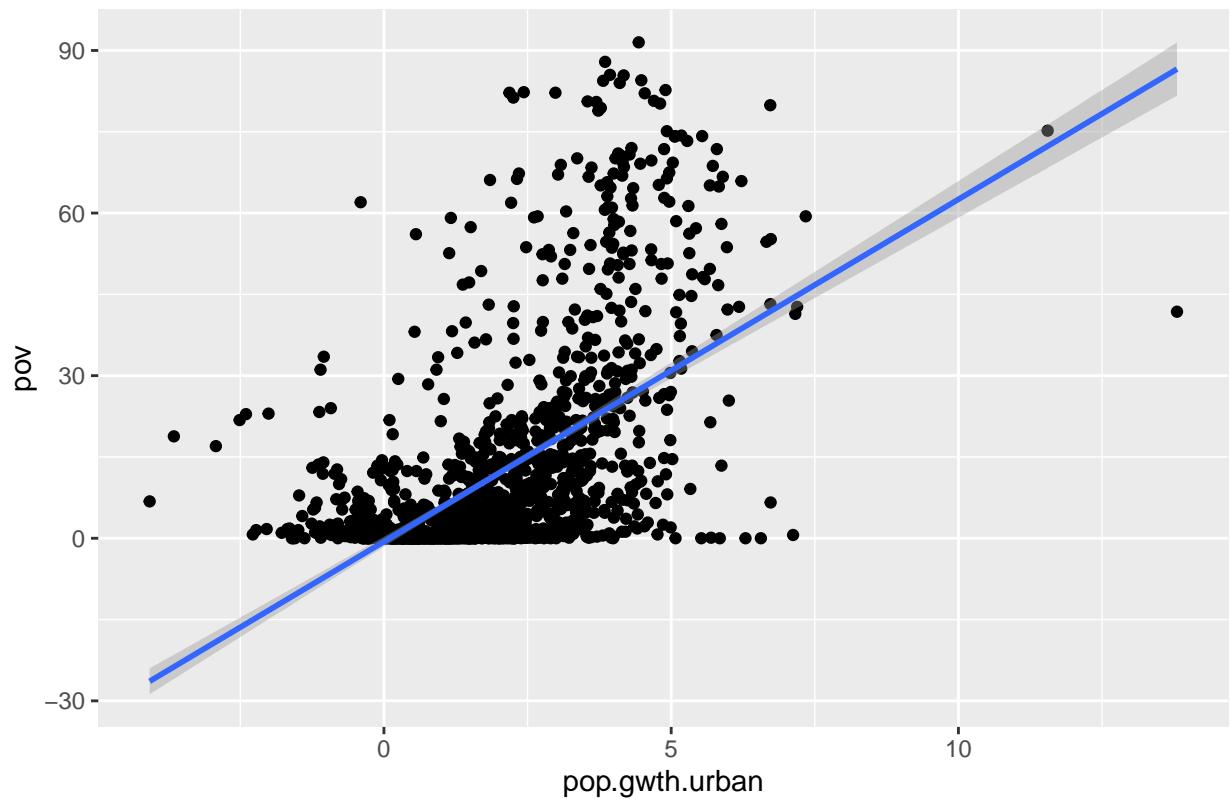
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{pop.gwth.rural}$



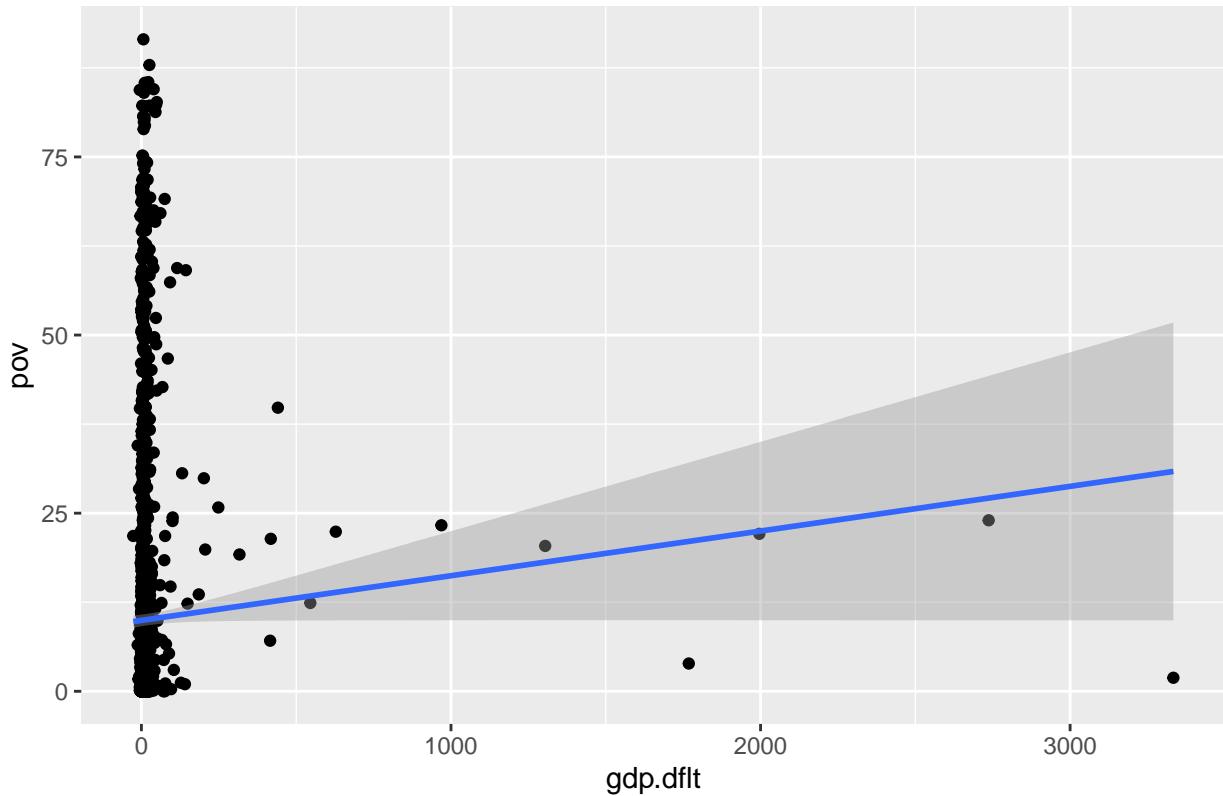
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{pop.gwth.urban}$



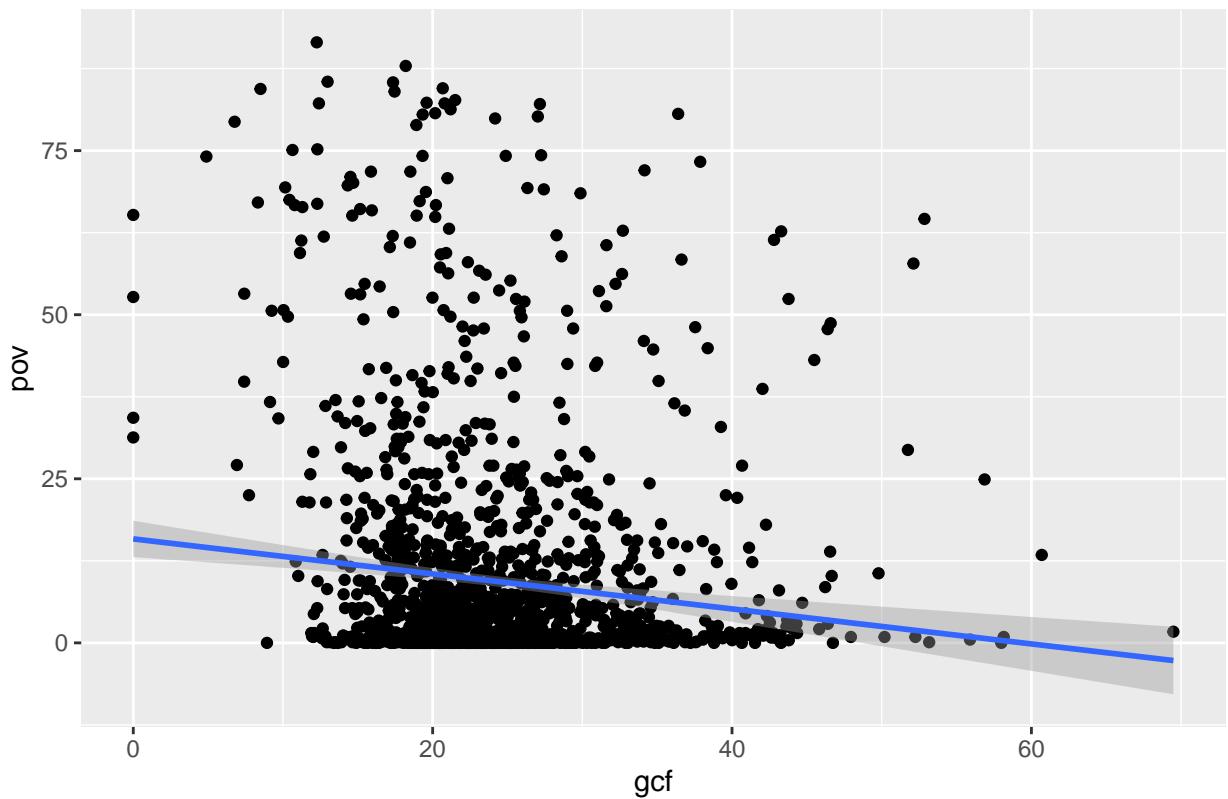
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{gdp.dflt}$



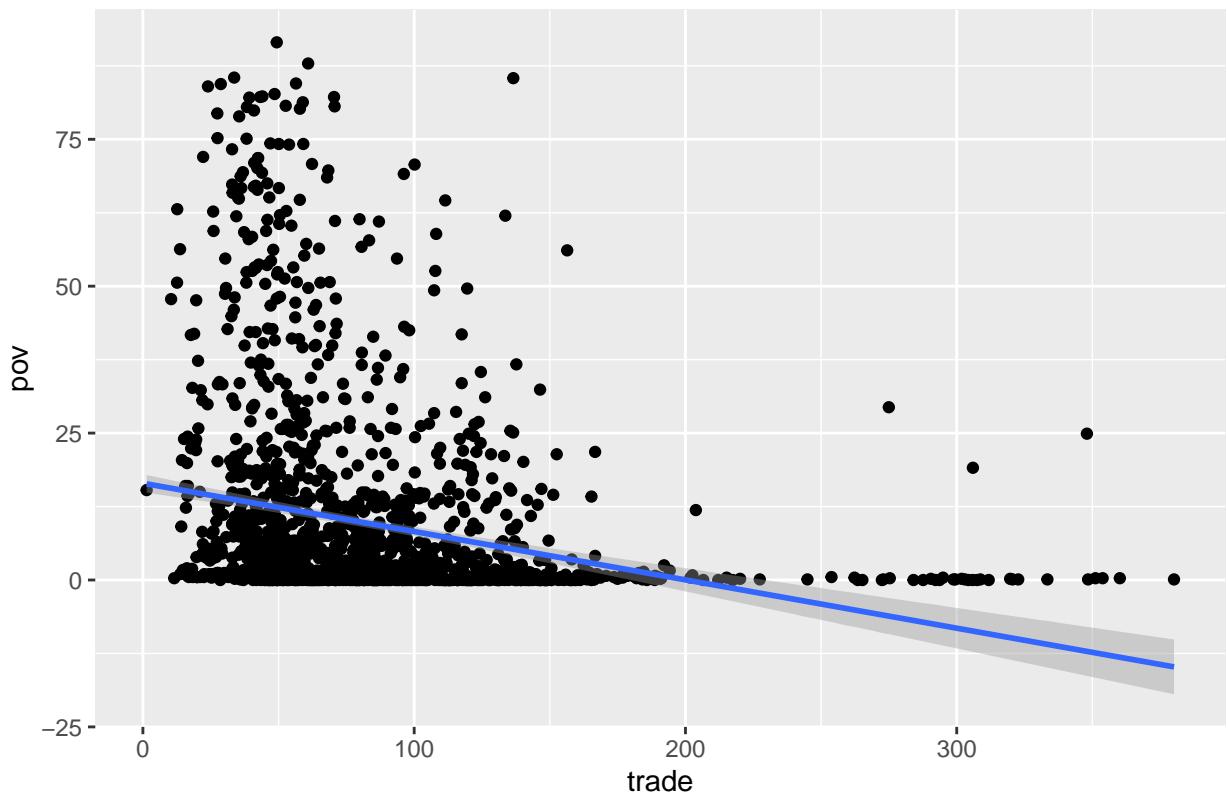
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{gcf}$



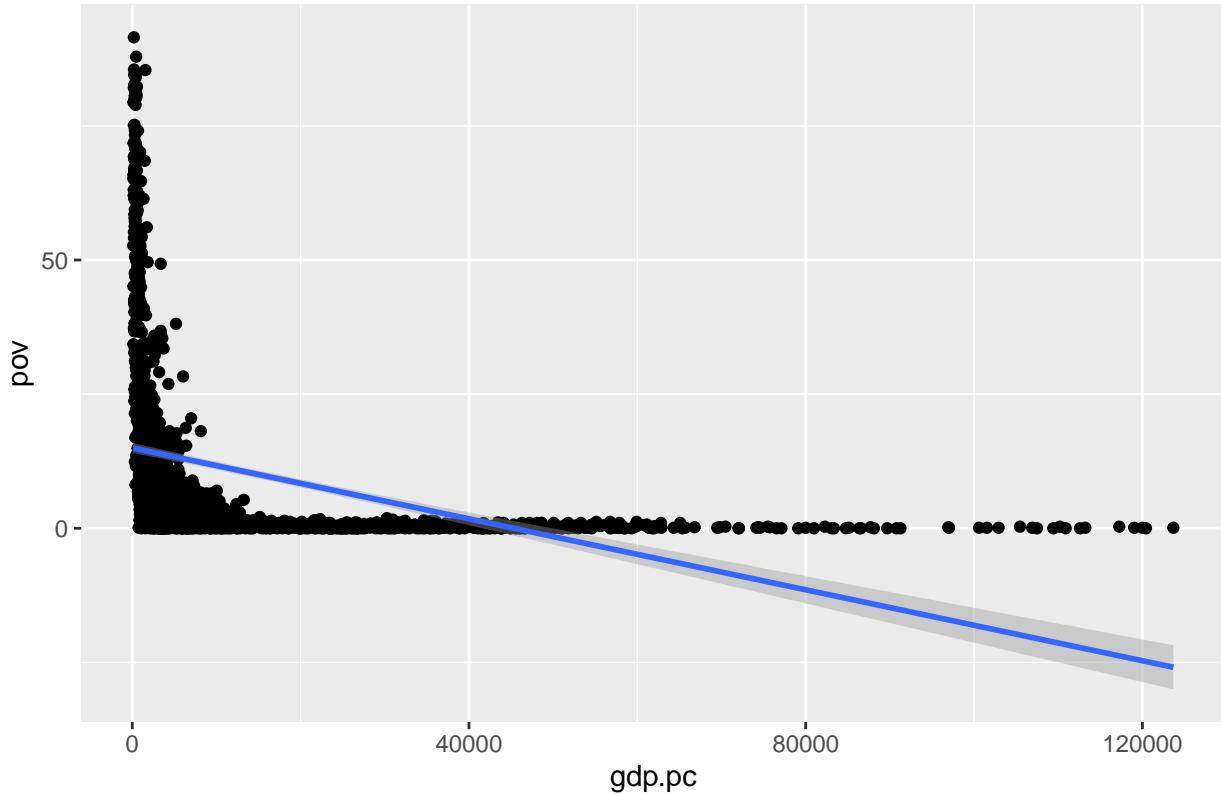
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{trade}$



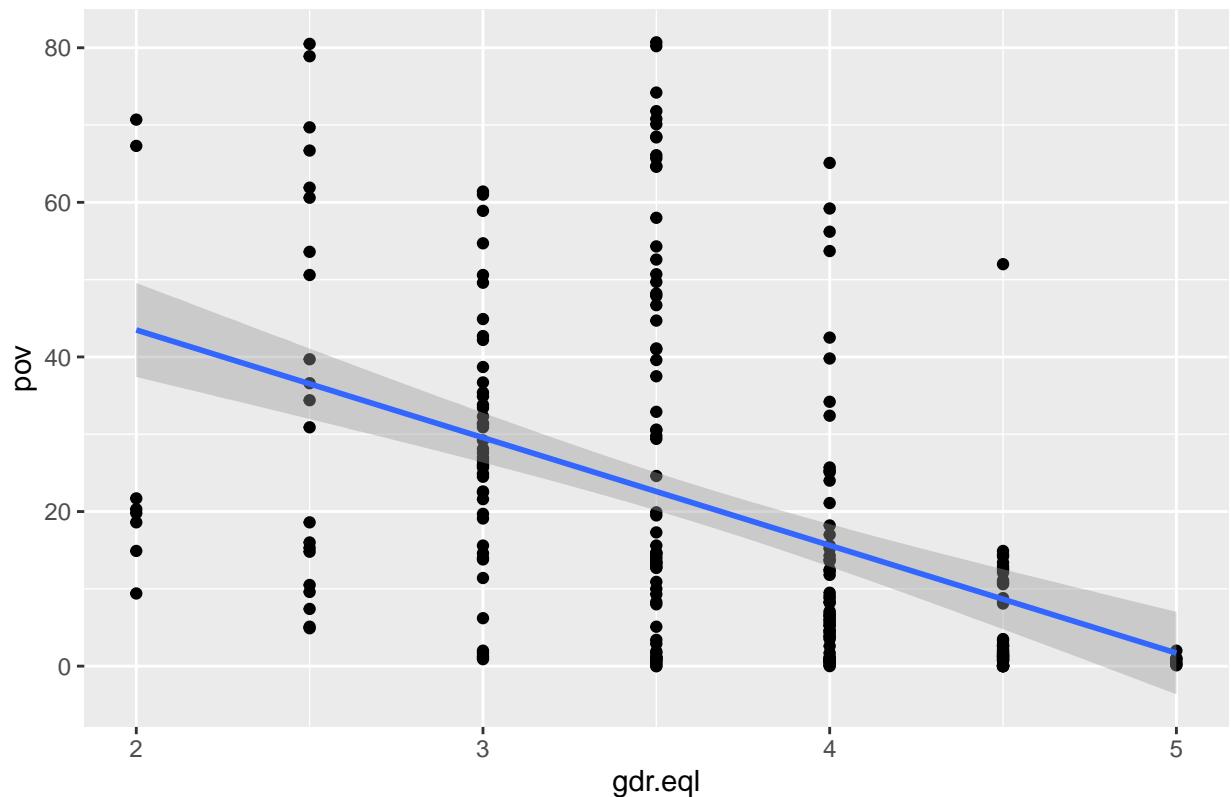
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{gdp.pc}$



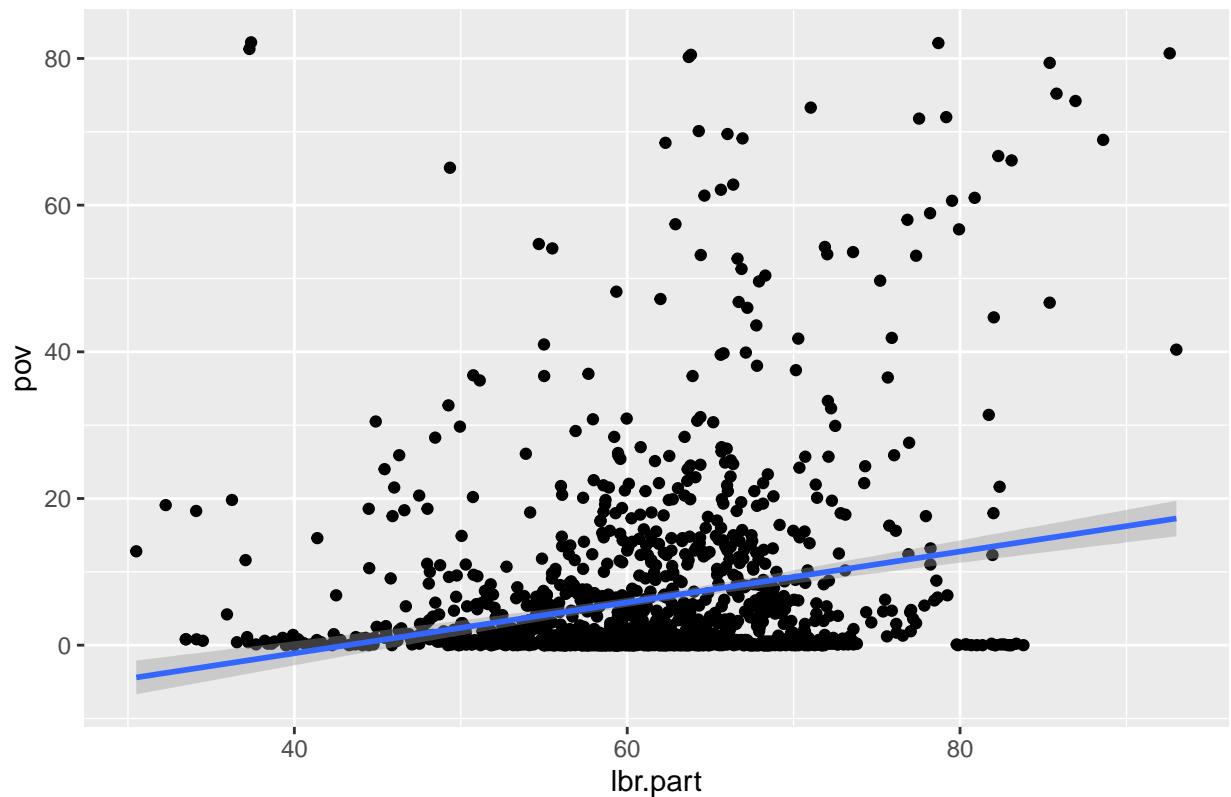
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{gdr.eq|}$



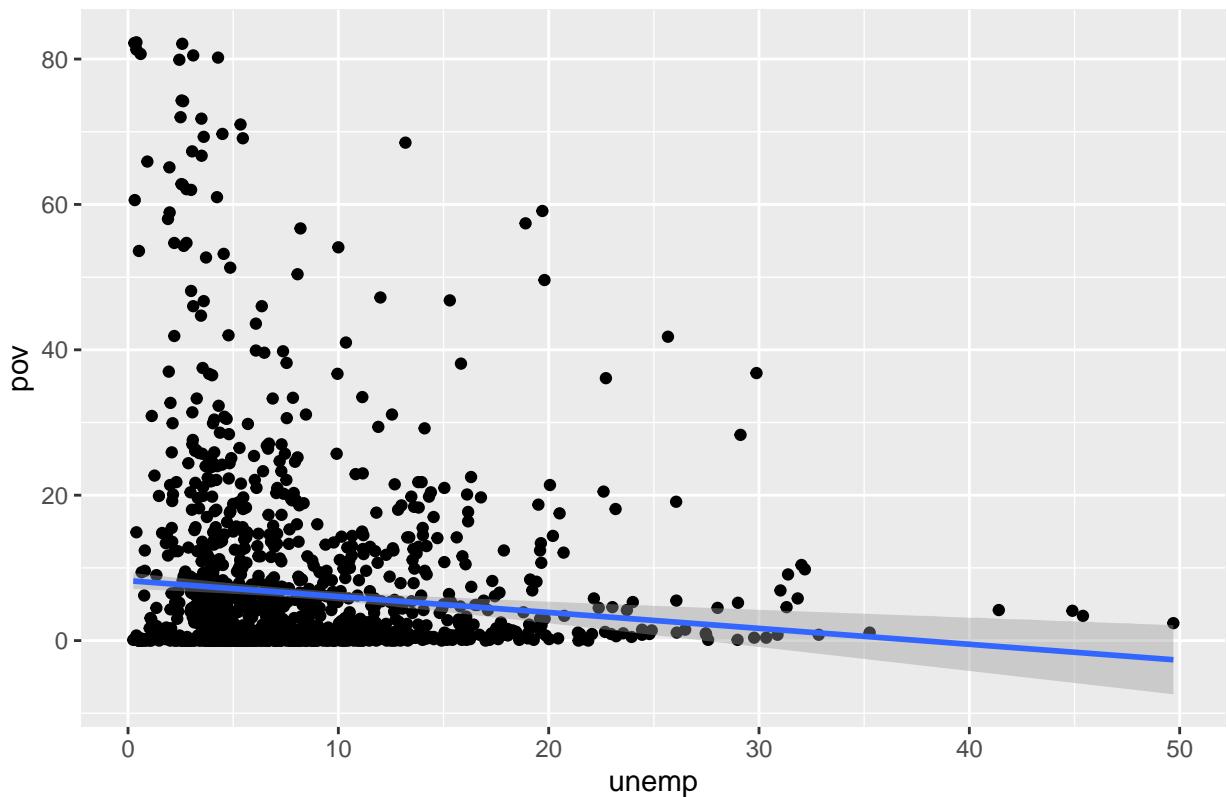
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{lbr.part}$



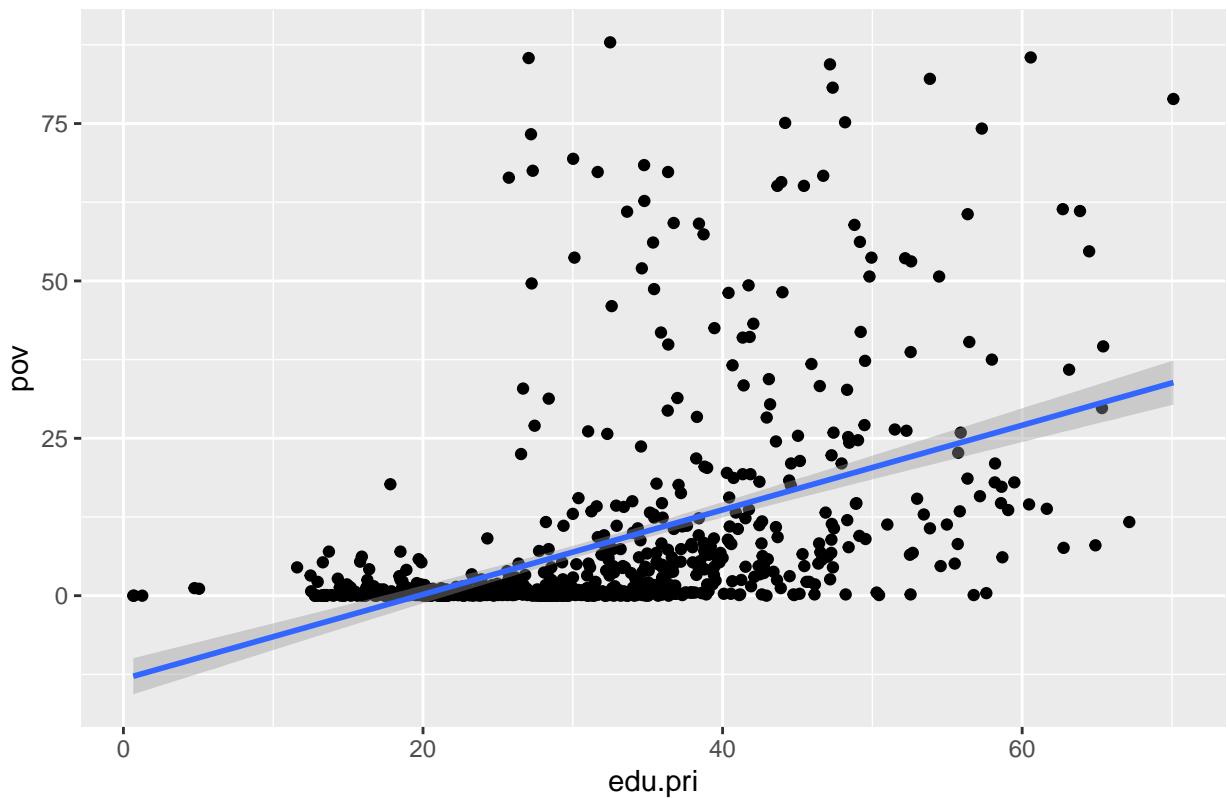
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{unemp}$



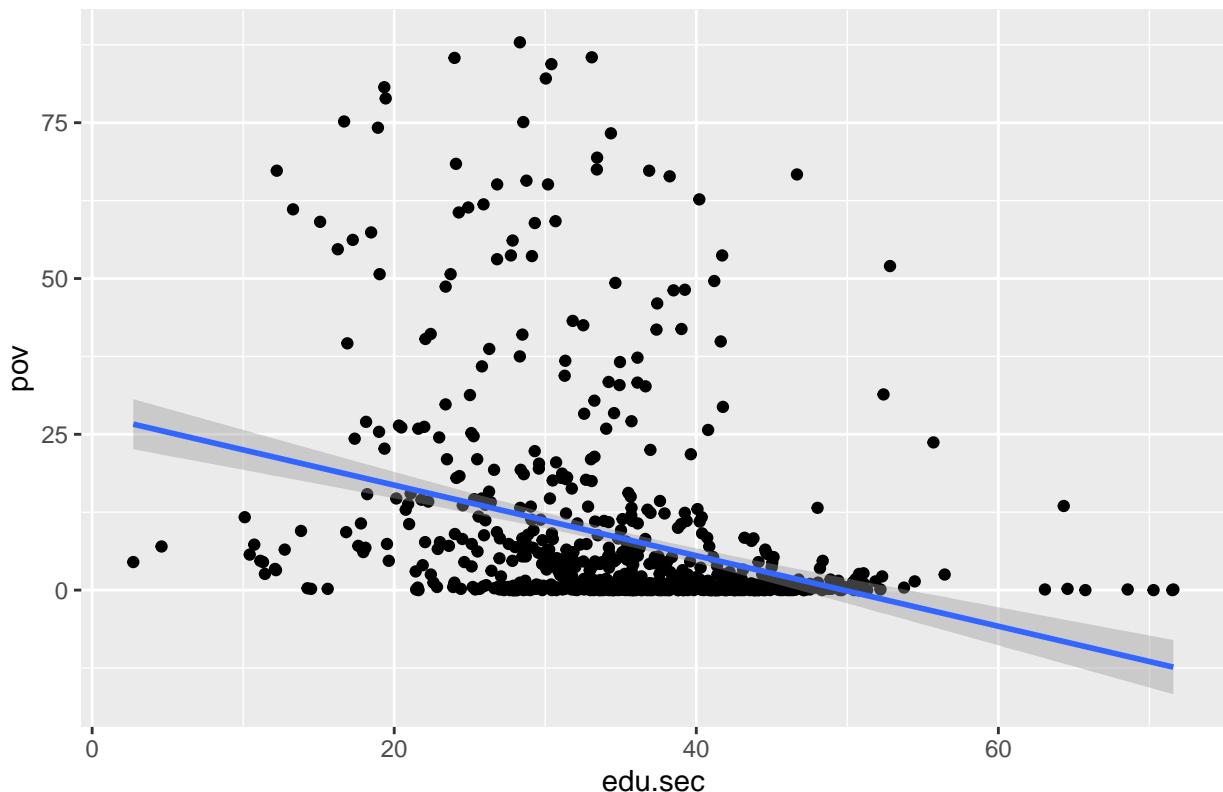
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{edu.pri}$



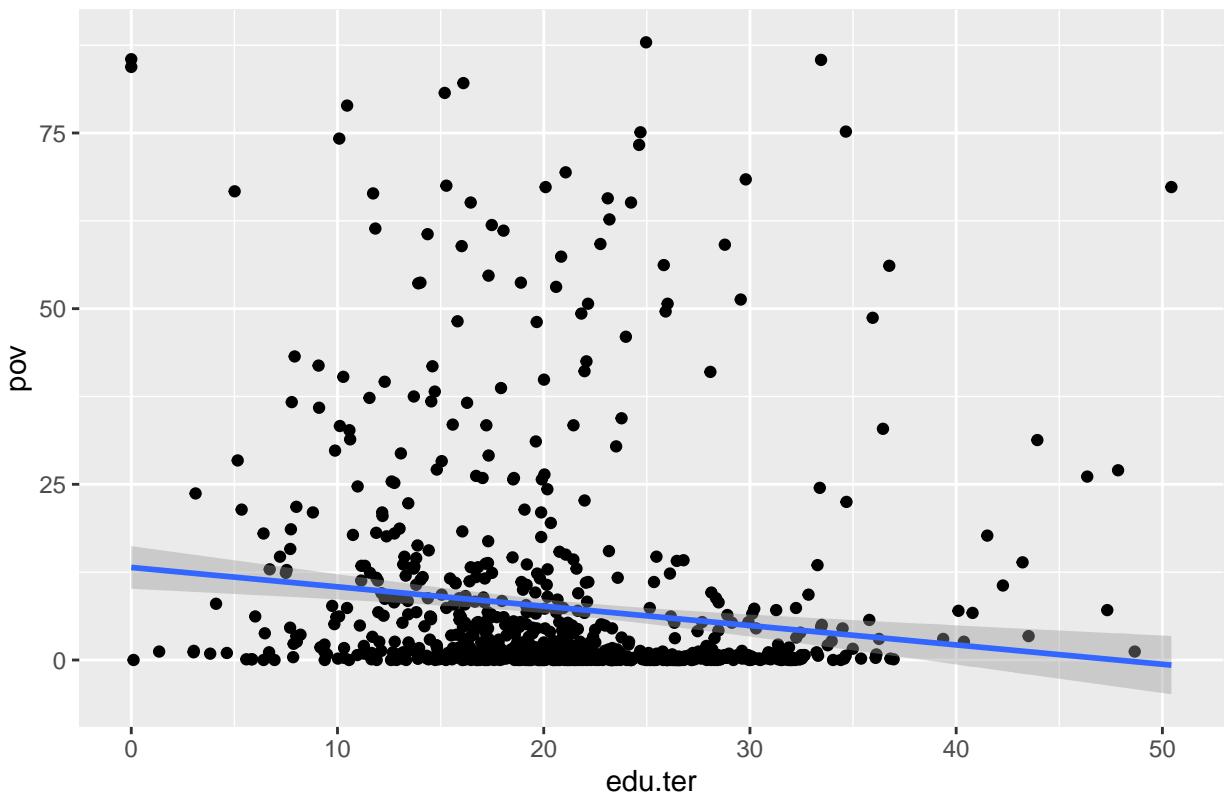
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{edu.sec}$



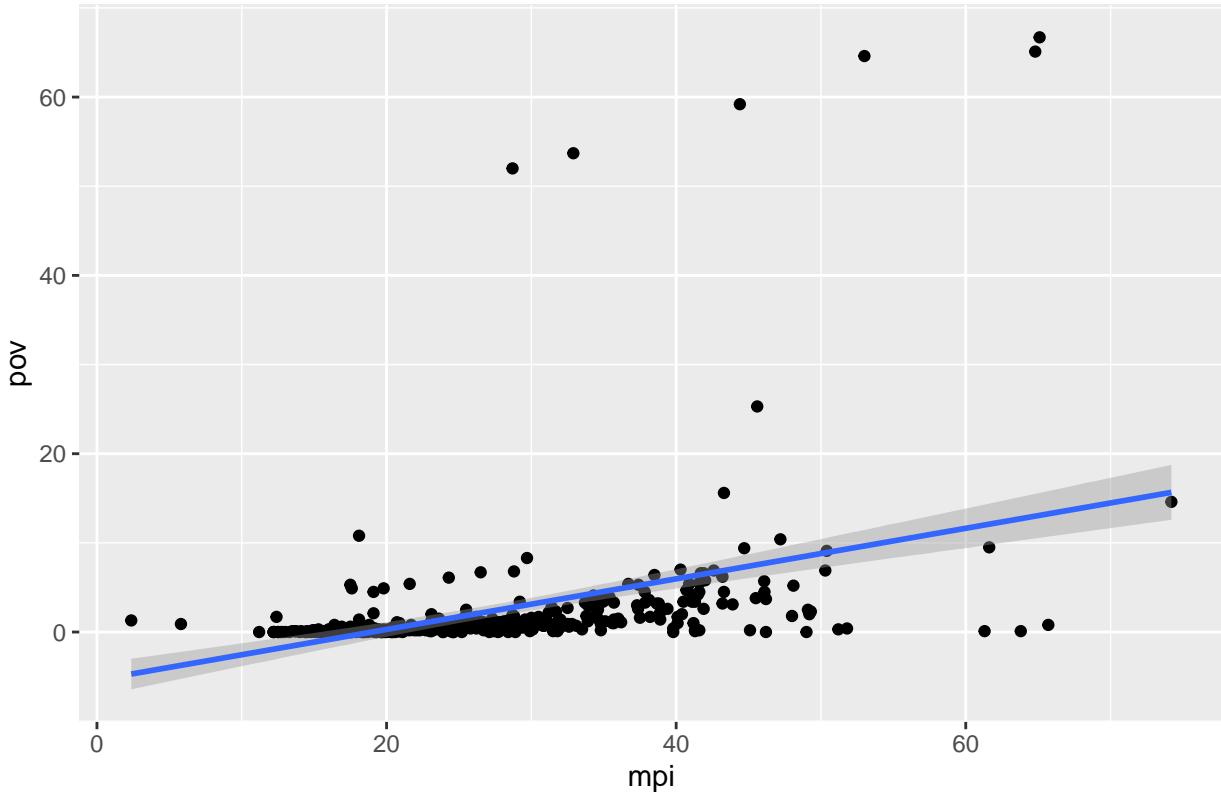
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \text{edu.ter}$



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

Relationship pov ~ mpi



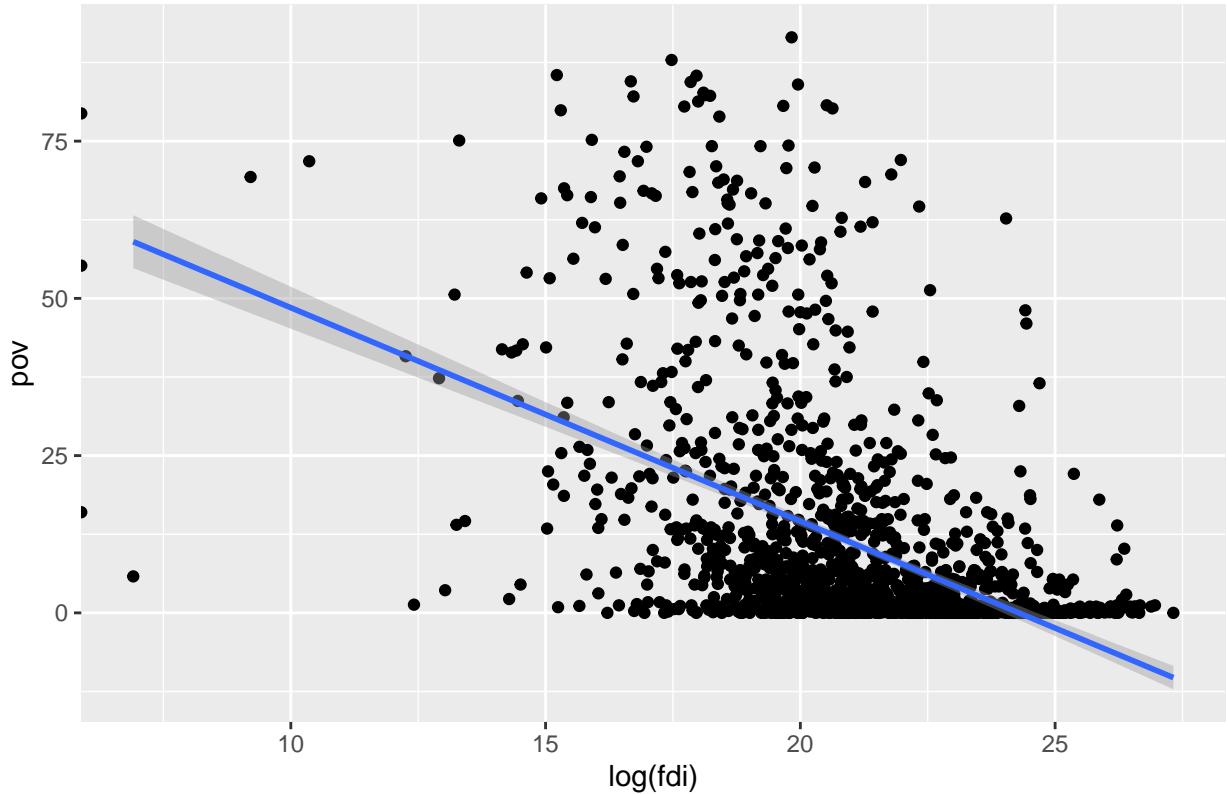
Most variables display linear relationship with some exceptions, which are more appropriate to assume a logarithmic relationship. Gender equality should be viewed as a categorical data.

```
logIvs <- c("fdi", "gdp.dflt", "gdp.pc")

for (indvar in logIvs) {
  print(ggplot(countries.num2 %>%
    filter(variable == indvar), aes(x = log(value),
    y = pov)) + geom_point() + geom_smooth(method = lm) +
    labs(title = paste0("Relationship pov ~ log(",
      indvar, ")"), x = paste0("log(", indvar,
      ")"), y = "pov"))
}

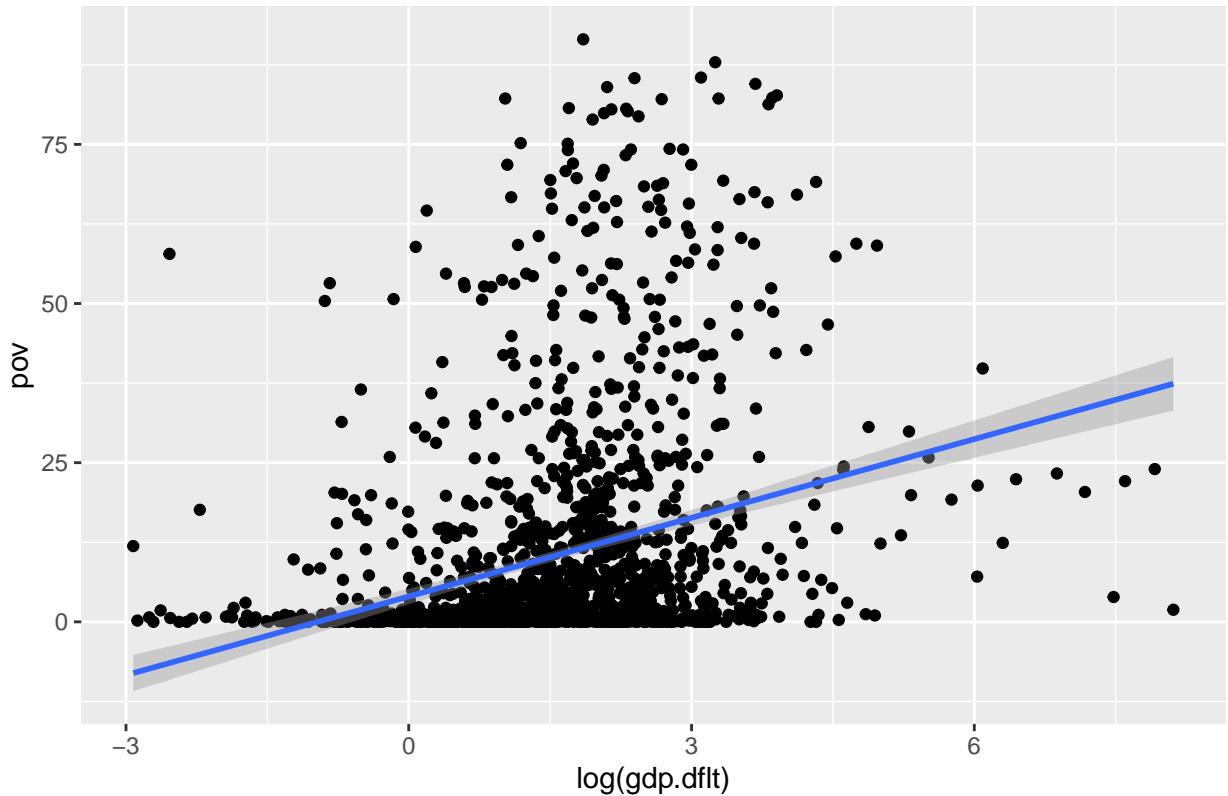
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \log(\text{fdi})$



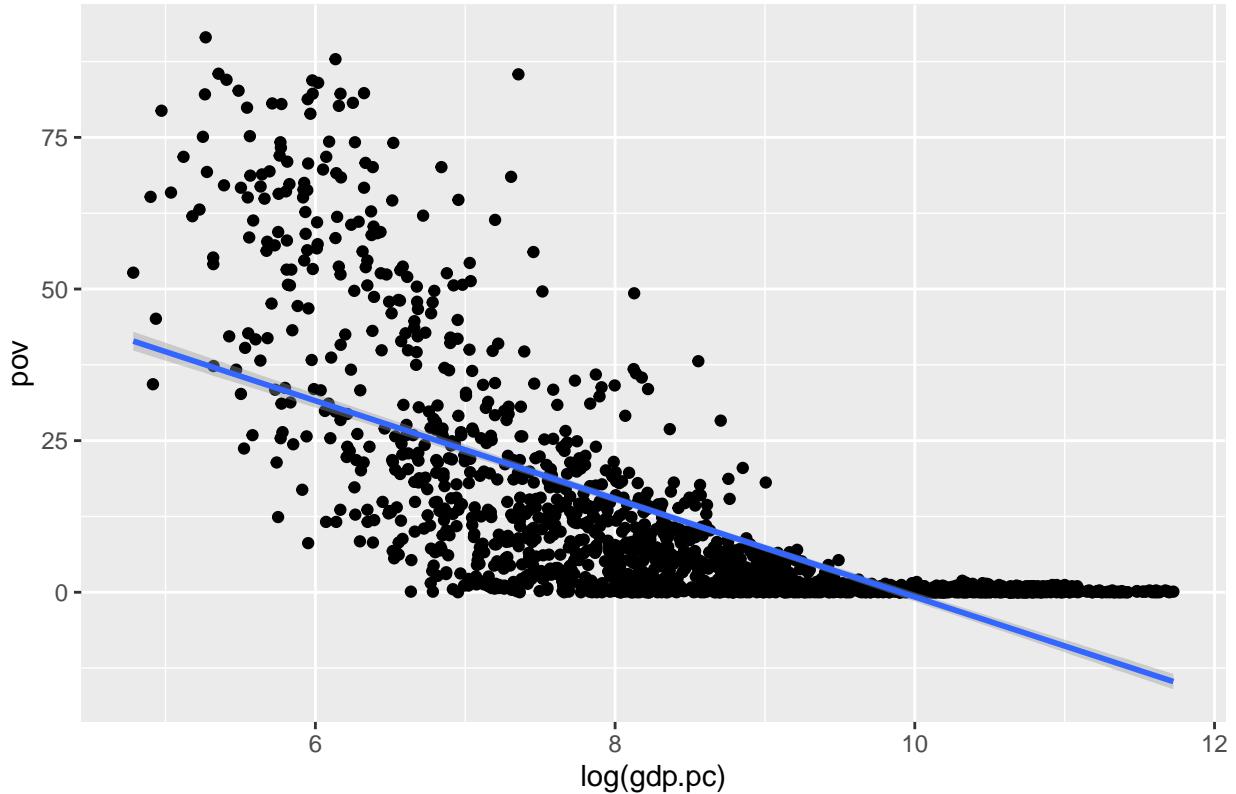
```
## `geom_smooth()` using formula 'y ~ x'
```

Relationship $\text{pov} \sim \log(\text{gdp.dflt})$



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

Relationship $pov \sim \log(gdp.pc)$



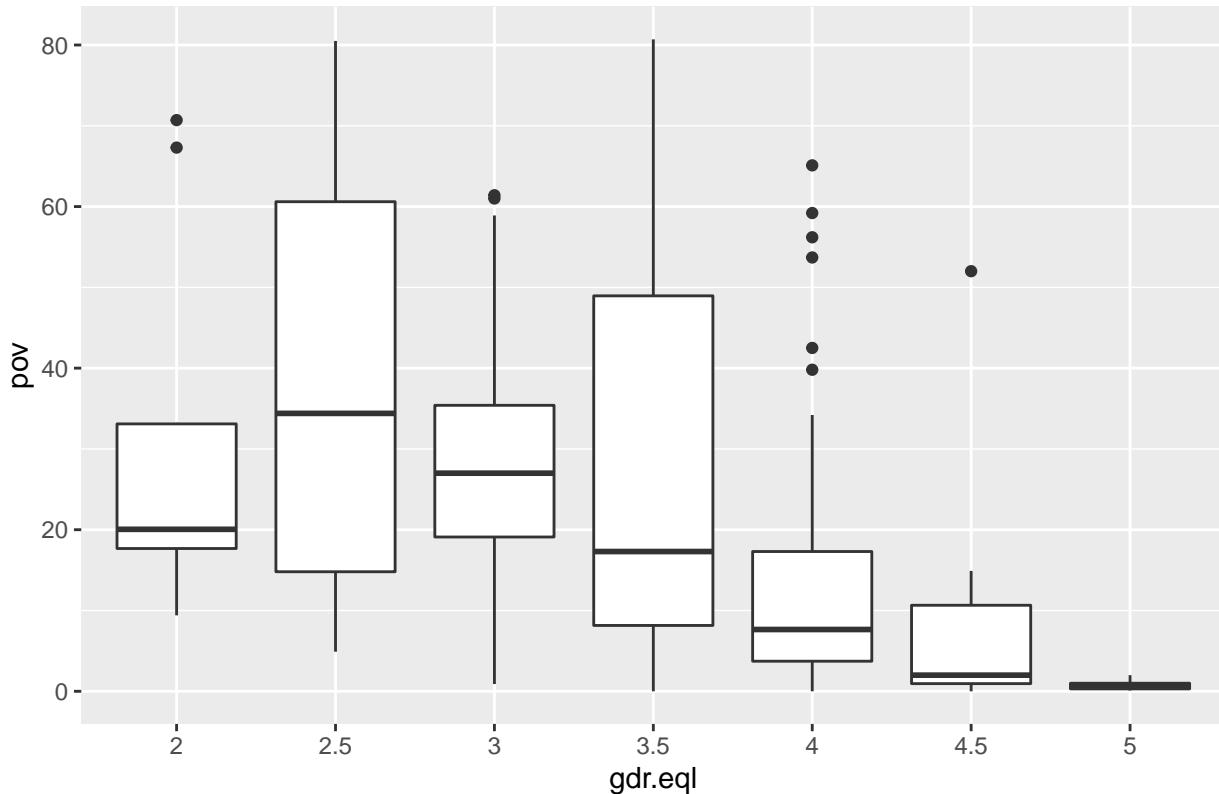
The relationship appears more linear after the transformation.

```
gdr.eql <- countries.num2 %>%
  filter(variable == "gdr.eql") %>%
  mutate(value = factor(value))

ggplot(gdr.eql, aes(x = value, y = pov)) + geom_boxplot() +
  geom_smooth(method = lm) + labs(title = paste("Relationship pov ~ gdr.eql"),
  x = "gdr.eql", y = "pov")

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

Relationship pov ~ gdr.eql



There's a significant correlation between gender equality and poverty.
Correlation coefficients between pov and predictors.

```
# transform some variables
countries.num3 <- countries.num %>%
  mutate(lfdi = log(fdi), lgdp.dflt = log(gdp.dflt),
    lgdp.pc = log(gdp.pc)) %>%
  mutate(lfdi = ifelse(is.nan(lfdi) | lfdi == -Inf,
    NA, lfdi))

# name of the independent variables
cn <- colnames(countries.num3)[-c(1, 2)]

# correlation of independent variable and pov
corWithPov <- function(indevar, data) {
  cor(data[[indevar]], data[["pov"]], use = "complete.obs")
}

cor.pov <- sapply(cn, corWithPov, data = countries.num3)
kable(cor.pov)
```

| | x |
|-----------|------------|
| mpi | 0.4036012 |
| edu.total | -0.3098112 |
| edu.pri | 0.4711269 |

| | x |
|----------------|------------|
| edu.sec | -0.3172901 |
| edu.ter | -0.1284782 |
| hlth | -0.2938428 |
| mil | -0.0069375 |
| fdi | -0.1441670 |
| lbr.part | 0.2316754 |
| unemp | -0.0947166 |
| pop.gwth.total | 0.5290581 |
| pop.gwth.rural | 0.4943156 |
| pop.gwth.urban | 0.5866539 |
| gdp.dflt | 0.0457111 |
| gdr.eql | -0.4505635 |
| gcf | -0.1108507 |
| trade | -0.2387164 |
| gdp.pc | -0.3773475 |
| lfdi | -0.4875988 |
| lgdp.dflt | 0.2992969 |
| lgdp.pc | -0.7053998 |

edu.ter, mil, fdi, lbr.part, unemp, gdp.dflt, gcf have negligible correlation with pov. lgdp.pc, lfdi, and lgdp.dflt have stronger linear relationship with pov than their un-transformed counterparts.

```
countries1 <- countries1 %>%
  mutate(lfdi = log(fdi), lgdp.dflt = log(gdp.dflt),
        lgdp.pc = log(gdp.pc)) %>%
  mutate(lfdi = ifelse(is.nan(lfdi) | lfdi == -Inf,
                      NA, lfdi))
```

2. Predictors are independent There should be no correlation/multicollinearity between each pair of predictors.

```
countries.arr <- simplify2array(countries.num %>%
  select(-c("year", "pov")))

cor mtx <- rcorr(countries.arr, type = "spearman")

round(cor mtx$r, 2)
```

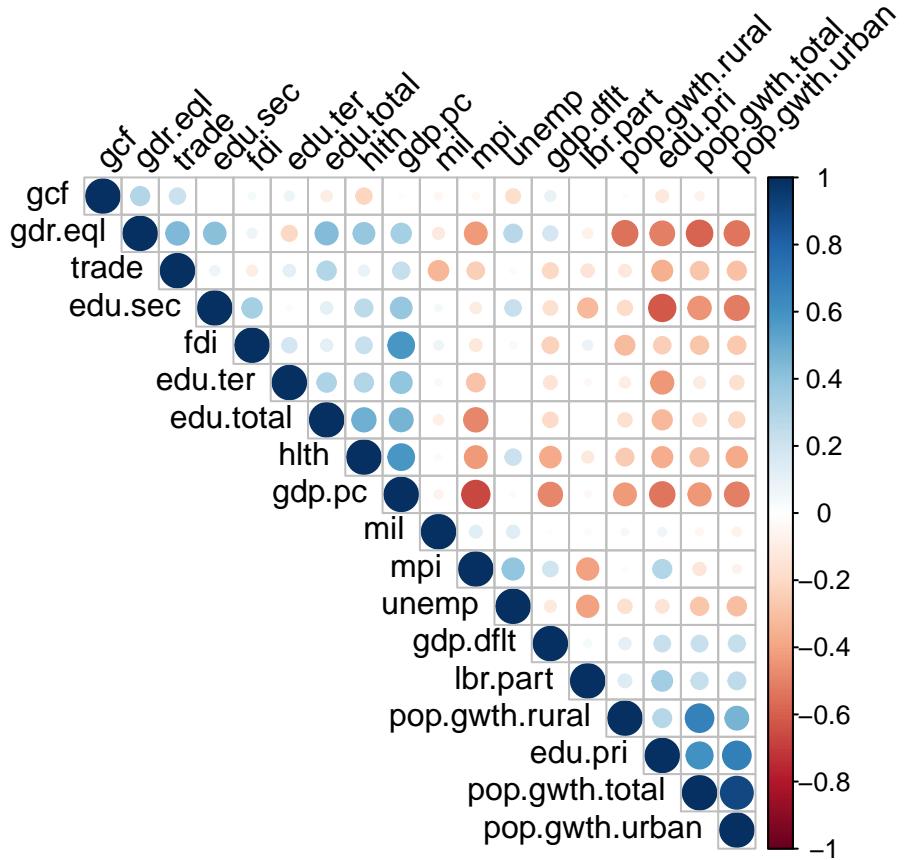
| | | | | | | | | |
|-------------------|-------|-----------|---------|---------|---------|-------|-------|-------|
| ## | mpi | edu.total | edu.pri | edu.sec | edu.ter | hlth | mil | fdi |
| ## mpi | 1.00 | -0.49 | 0.29 | -0.10 | -0.28 | -0.43 | 0.13 | -0.13 |
| ## edu.total | -0.49 | 1.00 | -0.33 | 0.12 | 0.31 | 0.48 | -0.08 | 0.12 |
| ## edu.pri | 0.29 | -0.33 | 1.00 | -0.62 | -0.44 | -0.36 | 0.06 | -0.24 |
| ## edu.sec | -0.10 | 0.12 | -0.62 | 1.00 | 0.02 | 0.26 | 0.04 | 0.34 |
| ## edu.ter | -0.28 | 0.31 | -0.44 | 0.02 | 1.00 | 0.30 | 0.04 | 0.19 |
| ## hlth | -0.43 | 0.48 | -0.36 | 0.26 | 0.30 | 1.00 | -0.03 | 0.22 |
| ## mil | 0.13 | -0.08 | 0.06 | 0.04 | 0.04 | -0.03 | 1.00 | 0.07 |
| ## fdi | -0.13 | 0.12 | -0.24 | 0.34 | 0.19 | 0.22 | 0.07 | 1.00 |
| ## lbr.part | -0.41 | 0.01 | 0.35 | -0.33 | -0.04 | -0.11 | 0.02 | 0.08 |
| ## unemp | 0.40 | 0.00 | -0.14 | 0.22 | 0.00 | 0.22 | 0.14 | -0.04 |
| ## pop.gwth.total | -0.14 | -0.14 | 0.60 | -0.45 | -0.11 | -0.29 | -0.05 | -0.27 |
| ## pop.gwth.rural | -0.02 | -0.16 | 0.29 | -0.18 | -0.10 | -0.26 | 0.04 | -0.32 |

```

## pop.gwth.urban -0.06    -0.21    0.68   -0.51   -0.16  -0.37  -0.07  -0.27
## gdp.dflt      0.19     -0.19    0.22   -0.17   -0.15  -0.38   0.01  -0.22
## gdr.eql      -0.43     0.44   -0.50    0.41   -0.21   0.39  -0.11   0.08
## gcf        -0.04    -0.10   -0.12    0.01    0.08  -0.21  -0.05   0.04
## trade       -0.24     0.30   -0.35    0.08    0.12   0.09  -0.34  -0.09
## gdp.pc       -0.66     0.47   -0.53    0.38    0.39   0.59  -0.06   0.59
##          lbr.part unemp pop.gwth.total pop.gwth.rural pop.gwth.urban
## mpi           -0.41    0.40    -0.14   -0.02   -0.06
## edu.total     0.01    0.00    -0.14   -0.16   -0.21
## edu.pri       0.35   -0.14    0.60    0.29    0.68
## edu.sec      -0.33    0.22   -0.45   -0.18   -0.51
## edu.ter      -0.04    0.00   -0.11   -0.10   -0.16
## hlth         -0.11    0.22   -0.29   -0.26   -0.37
## mil          0.02    0.14   -0.05    0.04   -0.07
## fdi          0.08   -0.04   -0.27   -0.32   -0.27
## lbr.part      1.00   -0.41    0.24    0.15    0.25
## unemp        -0.41    1.00   -0.27   -0.16   -0.30
## pop.gwth.total 0.24   -0.27    1.00    0.68    0.91
## pop.gwth.rural 0.15   -0.16    0.68    1.00    0.46
## pop.gwth.urban 0.25   -0.30    0.91    0.46    1.00
## gdp.dflt      0.05   -0.11    0.22    0.11    0.23
## gdr.eql      -0.08    0.27   -0.58   -0.55   -0.53
## gcf          0.01   -0.17   -0.06    0.01    0.00
## trade        -0.14   -0.03   -0.28   -0.13   -0.29
## gdp.pc        -0.04    0.02   -0.44   -0.43   -0.50
##          gdp.dflt gdr.eql   gcf trade gdp.pc
## mpi           0.19   -0.43  -0.04  -0.24   -0.66
## edu.total     -0.19    0.44  -0.10  0.30    0.47
## edu.pri       0.22   -0.50  -0.12  -0.35   -0.53
## edu.sec      -0.17    0.41   0.01   0.08    0.38
## edu.ter      -0.15   -0.21   0.08   0.12    0.39
## hlth         -0.38    0.39  -0.21   0.09    0.59
## mil          0.01   -0.11  -0.05  -0.34   -0.06
## fdi          -0.22    0.08   0.04  -0.09    0.59
## lbr.part      0.05   -0.08   0.01  -0.14   -0.04
## unemp        -0.11    0.27  -0.17  -0.03    0.02
## pop.gwth.total 0.22   -0.58  -0.06  -0.28   -0.44
## pop.gwth.rural 0.11   -0.55  0.01  -0.13   -0.43
## pop.gwth.urban 0.23   -0.53  0.00  -0.29   -0.50
## gdp.dflt      1.00    0.18   0.09  -0.20   -0.49
## gdr.eql      0.18    1.00   0.30   0.45    0.33
## gcf          0.09    0.30   1.00   0.21   -0.01
## trade        -0.20    0.45   0.21   1.00    0.24
## gdp.pc        -0.49    0.33  -0.01   0.24    1.00

corrplot(corr mtx$r, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)

```



Some significant correlations can be found between `gdr.eql` with `edu.pri`, `pop.gwth.total`, and `fdi` with `mil`, etc. We expect these variables to be eliminated in the VIF test. This might be a good property for imputation (3.2.4)

3.2. Ordinary Multiple Linear Regression

We conduct a normal linear regression, following the approaches mentioned above to address missing values issues.

3.2.1. Use Complete Cases (To be done)

3.2.1.1. Model Fitting

3.2.1.2. Assessment

3.2.1.3. Interpretation

3.2.2. Selectively remove variables with high missing rate (To be done)

3.2.2.1. Model Fitting

3.2.2.2. Assessment

3.2.2.3. Interpretation

3.2.3. Update the data set as we select variables (To be done)

3.2.3.1. Model Fitting

3.2.3.2. Assessment

3.2.3.3. Interpretation

3.2.4. Imputation

1. What is imputation

Imputation is a technique to handle missing values by replacing missing data with substitute values. In our study, we focus on “item imputation”, which means substituting for a component of a data point (i.e. a variable). The general idea is to take a value that preserve the property of the data (e.g., distribution, mean, standard deviation), or predict the value using other variables.

We use **Multivariate Imputation By Chained Equations (MICE)** as our primary tool for this technique, because of its wide acceptance in scientific studies (Alruhaymi and Kim (2021)). In order to use this technique, we have to assume that the “missingness” of a field can be explained by the values in other columns, (e.g., If the countries is in North America, it’s more likely to be missing as we have discovered in 2.2). The general ideas of the algorithm is iteratively predict the missing values based on other variables, and improve it after each iterations until predicted value converse to a stable point. The algorithmic details of MICE is very concisely explained in Gopalan (2020).

Our parameter for this algorithm is fairly standard:

- `m = 5` imputed data sets
- `maxit = 30` iterations

The imputation method is `cart` (Classification and Regression Trees).

```
set.seed(1984)
# split data
isComplete <- which(complete.cases(countries1))
idx <- sample(isComplete, replace = F, 0.3 * nrow(countries1))
countries.train.4 <- countries1[-idx, ]
countries.test.4 <- countries1[idx, ]
```

```
# blank for no imputation, cart for variable
# requiring imputation
meth <- c(rep("", 4), "cart", "", rep("cart", 16))
names(meth) <- colnames(countries1)
meth
```

```
##   country.code   country.name       year      pov      income
##           ""          ""        ""      ""      "cart"
##       reg      edu.total      hlth      mil      fdi
##       ""        "cart"      "cart"      "cart"      "cart"
##   lbr.part      unemp pop.gwth.total pop.gwth.rural pop.gwth.urban
```

```

##      "cart"      "cart"      "cart"      "cart"      "cart"
##      gdp.dflt      gcf      trade      gdp.pc      lfdi
##      "cart"      "cart"      "cart"      "cart"      "cart"
##      lgdp.dflt    lgdp.pc
##      "cart"      "cart"

# run the algorithm countries1.imputed <-
# mice(countries1, m = 3, maxit = 20, method =
# meth) saveRDS(countries1.imputed,
# 'countries1.imputed.RData')
countries1.imputed <- readRDS("countries1.imputed.RData")
summary(countries1.imputed)

## Class: mids
## Number of multiple imputations: 3
## Imputation methods:
##   country.code  country.name      year      pov      income
##      ""          ""          ""          ""          ""
##      reg        edu.total      hlth      mil      fdi
##      ""          "cart"      "cart"      "cart"      "cart"
##      lbr.part    unemp pop.gwth.total pop.gwth.rural pop.gwth.urban
##      "cart"      "cart"      ""          "cart"      "cart"
##      gdp.dflt      gcf      trade      gdp.pc      lfdi
##      "cart"      "cart"      "cart"      "cart"      "cart"
##      lgdp.dflt    lgdp.pc
##      "cart"      "cart"

## PredictorMatrix:
##   country.code country.name year pov income reg edu.total hlth mil
##   country.code      0          0  1  1  1  1  1  1  1
##   country.name      1          0  1  1  1  1  1  1  1
##   year             1          0  0  1  1  1  1  1  1
##   pov              1          0  1  0  1  1  1  1  1
##   income           1          0  1  1  0  1  1  1  1
##   reg              1          0  1  1  1  0  1  1  1
##   fdi lbr.part    unemp pop.gwth.total pop.gwth.rural pop.gwth.urban
##   country.code     1          1  1          1          1          1
##   country.name     1          1  1          1          1          1
##   year             1          1  1          1          1          1
##   pov              1          1  1          1          1          1
##   income           1          1  1          1          1          1
##   reg              1          1  1          1          1          1
##   gdp.dflt      gcf      trade      gdp.pc      lfdi
##   country.code     1          1  1          1          1          1
##   country.name     1          1  1          1          1          1
##   year             1          1  1          1          1          1
##   pov              1          1  1          1          1          1
##   income           1          1  1          1          1          1
##   reg              1          1  1          1          1          1

## Number of logged events: 961
##   it im      dep      meth
## 1  0  0      constant
## 2  1  1      income      cart
## 3  1  1      edu.total    cart
## 4  1  1      hlth       cart

```

```

## 5 1 1      mil      cart
## 6 1 1      fdi      cart
##
## 1
## 2
## 3          country.codeARE, country.codeBIH, country.codeCOD, country.codeDZA, country.codeEAS, co
## 4
## 5 country.codeBTN, country.codeCOM, country.codeDJI, country.codeEAS, country.codeECS, country.codeF
## 6

```

We can take a look into one of the imputed data sets.

```

countries1.imputed.1 <- complete(countries1.imputed,
  1)
summary(countries1.imputed.1)

##   country.code  country.name           year       pov      income
##   BRA      : 36  Length:1843      Min.  :1967  Min.  : 0.00 H :623
##   CRI      : 34  Class  :character  1st Qu.:2001  1st Qu.: 0.20 L :261
##   ARG      : 32  Mode   :character  Median :2008  Median : 1.50 LM:509
##   USA      : 32                   Mean   :2007  Mean   :10.04 UM:450
##   HND      : 30                   3rd Qu.:2014  3rd Qu.:11.60
##   GBR      : 29                   Max.   :2021  Max.   :91.50
##   (Other):1650
##
##             reg      edu.total       hlth
##   East Asia & Pacific    :167  Min.  : 1.033  Min.  : 1.718
##   Europe & Central Asia   :845  1st Qu.: 3.437  1st Qu.: 4.980
##   Latin America & Caribbean:416  Median : 4.388  Median : 6.771
##   Middle East & North Africa:104  Mean   : 4.482  Mean   : 6.813
##   North America            : 50  3rd Qu.: 5.376  3rd Qu.: 8.463
##   South Asia               : 53  Max.   :15.750  Max.   :17.733
##   Sub-Saharan Africa       :208
##
##      mil      fdi      lbr.part      unemp
##   Min.  : 0.000  Min.  :-3.444e+11  Min.  :30.50  Min.  : 0.250
##   1st Qu.: 1.040  1st Qu.: 2.982e+08  1st Qu.:56.31  1st Qu.: 4.415
##   Median : 1.465  Median : 1.702e+09  Median :61.46  Median : 6.770
##   Mean   : 1.784  Mean   : 1.613e+10  Mean   :61.14  Mean   : 8.083
##   3rd Qu.: 2.100  3rd Qu.: 9.793e+09  3rd Qu.:65.93  3rd Qu.:10.075
##   Max.   :19.385  Max.   : 7.338e+11  Max.   :93.00  Max.   :49.700
##
##   pop.gwth.total  pop.gwth.rural  pop.gwth.urban      gdp.dflt
##   Min.  :-3.6295  Min.  :-8.560655  Min.  :-4.078  Min.  : -26.300
##   1st Qu.: 0.2836  1st Qu.: -0.846827  1st Qu.: 0.517  1st Qu.:  1.734
##   Median : 1.0378  Median : -0.021085  Median : 1.485  Median :  3.890
##   Mean   : 1.0639  Mean   : 0.005301  Mean   : 1.692  Mean   : 16.129
##   3rd Qu.: 1.7764  3rd Qu.: 0.959723  3rd Qu.: 2.652  3rd Qu.:  8.606
##   Max.   : 5.6145  Max.   : 4.596858  Max.   :13.805  Max.   :3333.585
##
##      gcf      trade      gdp.pc      lfdi
##   Min.  : 0.00  Min.  : 1.378  Min.  : 119.7  Min.  : 6.908
##   1st Qu.:19.63  1st Qu.: 50.622  1st Qu.: 1904.4  1st Qu.:19.513
##   Median :22.71  Median : 72.606  Median : 5913.4  Median :21.255
##   Mean   :23.89  Mean   : 83.509  Mean   :14983.6  Mean   :20.767

```

```

## 3rd Qu.:26.78   3rd Qu.:104.706   3rd Qu.: 20863.0   3rd Qu.:23.008
## Max.    :69.48    Max.    :380.104    Max.    :123678.7   Max.    :27.322
##
##      lgdp.dflt          lgdp.pc
##  Min.   :-2.9224   Min.   : 4.785
##  1st Qu.: 0.5504   1st Qu.: 7.552
##  Median : 1.3584   Median : 8.685
##  Mean   : 1.2262   Mean   : 8.666
##  3rd Qu.: 2.1506   3rd Qu.: 9.946
##  Max.   : 8.1118   Max.   :11.725
##
## sum(!complete.cases(countries1.imputed.1))

## [1] 0

```

We found no missing cases as expected.

3.2.4.1. Model Fitting

A. Ordinary Linear Regression We generated 5 sets of imputed (training) data. We can build our model on each of these data sets and combine the estimates using [pooling rule](#). With all the generated data sets.

```

formula.str <- "pov ~ reg+year+income+edu.total+hlth+mil+fdi+lbr.part+unemp+pop.gwth.total+pop.gwth.rur

# Using only the first imputed data set
fit1 <- lm(as.formula(formula.str), data = complete(countries1.imputed,
  1))
summary(fit1)

##
## Call:
## lm(formula = as.formula(formula.str), data = complete(countries1.imputed,
##   1))
##
## Residuals:
##      Min       1Q     Median       3Q      Max
## -40.530  -3.070  -0.009   2.574  54.008
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.962e+02  4.793e+01   6.179 7.93e-10 ***
## regEurope & Central Asia -1.643e+00  8.692e-01  -1.891 0.058846 .
## regLatin America & Caribbean 5.615e-02  8.680e-01   0.065 0.948428
## regMiddle East & North Africa -9.028e-01  1.208e+00  -0.747 0.454985
## regNorth America            -4.258e+00  1.584e+00  -2.689 0.007242 **
## regSouth Asia               -6.935e-01  1.342e+00  -0.517 0.605483
## regSub-Saharan Africa       1.658e+01  1.021e+00  16.245 < 2e-16 ***
## year                         -1.056e-01  2.437e-02  -4.332 1.56e-05 ***
## incomeL                      -4.909e+00  1.866e+00  -2.631 0.008581 **

```

```

## incomeLM          -1.090e+01  1.207e+00 -9.027 < 2e-16 ***
## incomeUM          -6.180e+00  8.523e-01 -7.251 6.11e-13 ***
## edu.total         -7.838e-01  1.383e-01 -5.668 1.68e-08 ***
## hlth              3.293e-01  1.120e-01  2.939 0.003333 **
## mil               -7.389e-01  1.663e-01 -4.443 9.42e-06 ***
## fdi               6.119e-12  4.776e-12  1.281 0.200284
## lbr.part           2.008e-01  2.584e-02  7.772 1.29e-14 ***
## unemp              3.551e-02  3.940e-02  0.901 0.367623
## pop.gwth.total    -2.332e+00  5.959e-01 -3.914 9.42e-05 ***
## pop.gwth.rural    8.943e-01  2.360e-01  3.790 0.000156 ***
## pop.gwth.urban    2.048e+00  3.438e-01  5.958 3.07e-09 ***
## gdp.dflt           -7.882e-04  1.574e-03 -0.501 0.616586
## gcf               -1.128e-01  2.944e-02 -3.831 0.000132 ***
## trade              -1.437e-02  4.743e-03 -3.030 0.002478 **
## gdp.pc              2.323e-04  2.353e-05  9.872 < 2e-16 ***
## lfdi              1.315e-03  6.836e-02  0.019 0.984660
## lgdp.dflt           1.905e-01  1.481e-01  1.286 0.198547
## lgdp.pc             -9.332e+00  5.635e-01 -16.561 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.035 on 1816 degrees of freedom
## Multiple R-squared:  0.793, Adjusted R-squared:  0.79
## F-statistic: 267.6 on 26 and 1816 DF, p-value: < 2.2e-16

fit1.pred <- predict(fit1, countries.test.4)

# Build models with all generated data set and
# pool the estimates
fit2 <- with(data = countries1.imputed, exp = lm(as.formula(formula.str)))
fit2.combined <- pool(fit2)
summary(fit2.combined)

##                               term      estimate   std.error   statistic
## 1                   (Intercept) 2.977055e+02 5.237427e+01  5.6841927
## 2       regEurope & Central Asia -1.462788e+00 9.590422e-01 -1.5252592
## 3     regLatin America & Caribbean 3.094891e-01 1.022914e+00  0.3025562
## 4   regMiddle East & North Africa -1.050869e+00 1.243778e+00 -0.8449013
## 5       regNorth America -3.986980e+00 1.627780e+00 -2.4493365
## 6       regSouth Asia -4.342683e-01 1.463266e+00 -0.2967802
## 7     regSub-Saharan Africa 1.693675e+01 1.305242e+00 12.9759478
## 8                  year -1.092433e-01 2.701994e-02 -4.0430621
## 9                  incomeL -4.042467e+00 2.321582e+00 -1.7412555
## 10                 incomeLM -1.018739e+01 1.436036e+00 -7.0941036
## 11                 incomeUM -5.755437e+00 9.647468e-01 -5.9657490
## 12                 edu.total -6.892835e-01 1.743769e-01 -3.9528382
## 13                  hlth  2.941010e-01 1.219153e-01  2.4123385
## 14                  mil  -6.820828e-01 2.285260e-01 -2.9847049
## 15                  fdi  4.721357e-12 5.549491e-12  0.8507729
## 16                 lbr.part 1.984601e-01 5.561103e-02  3.5687176
## 17                  unemp 2.948381e-02 4.117469e-02  0.7160663
## 18     pop.gwth.total -2.323612e+00 6.669702e-01 -3.4838316
## 19     pop.gwth.rural  9.290061e-01 2.511417e-01  3.6991304
## 20     pop.gwth.urban  2.043763e+00 3.498012e-01  5.8426429

```

```

## 21          gdp.dflt -7.908469e-04 1.596442e-03 -0.4953808
## 22          gcf -1.024032e-01 3.469498e-02 -2.9515287
## 23          trade -1.268486e-02 5.666923e-03 -2.2384037
## 24          gdp.pc  2.247952e-04 2.795218e-05  8.0421343
## 25          lfdi   1.205035e-01 2.345661e-01  0.5137295
## 26          lgdp.dflt 1.635998e-01 1.530945e-01  1.0686195
## 27          lgdp.pc -9.070002e+00 6.248341e-01 -14.5158572
##           df      p.value
## 1    102.582658 1.241832e-07
## 2     72.130457 1.315674e-01
## 3     28.342586 7.644375e-01
## 4     592.448853 3.985070e-01
## 5    1036.088455 1.447670e-02
## 6     95.852317 7.672765e-01
## 7    14.302842 2.634816e-09
## 8     77.689786 1.232936e-04
## 9     15.838551 1.010236e-01
## 10    24.306394 2.298142e-07
## 11    48.803143 2.677759e-07
## 12    17.981512 9.344578e-04
## 13   132.994795 1.721462e-02
## 14    9.461645 1.452527e-02
## 15   30.736672 4.014749e-01
## 16    3.200287 3.384644e-02
## 17   602.700949 4.742278e-01
## 18    56.608308 9.609871e-04
## 19   174.545106 2.896391e-04
## 20   1168.602855 6.655287e-09
## 21   1692.244160 6.203957e-01
## 22   25.967472 6.623360e-03
## 23   24.568512 3.450592e-02
## 24   24.514312 2.463564e-08
## 25   2.778955 6.454736e-01
## 26   382.313448 2.859154e-01
## 27   58.156006 0.000000e+00

```

```

# dummy lm model
fit2.dummy <- lm(as.formula(formula.str), data = countries1)
# replace coefficients of dummy model & predict
fit2.dummy$coefficients <- fit2.combined$pooled$estimate
fit2.pred <- predict(fit2.dummy, countries.test.4)

```

Helper function to calculate R-Squared

```

r2 <- function(pred, orig) {
  RSS <- sum((pred - orig)^2)
  TSS <- sum((pred - mean(orig))^2)
  R2 <- 1 - RSS/TSS
  return(R2)
}

adjR2 <- function(pred, orig, k) {
  R2 <- r2(pred, orig)

```

```

    n <- length(pred)
    adjr2 <- 1 - (1 - R2) * (n - 1)/(n - k - 1)
    return(adjr2)
}

```

Adjusted R-Squared on train and test

```

k <- length(fit1$coefficients) - 1
fit1.train.r2 <- summary(fit1)$adj.r.squared
fit1.test.r2 <- adjR2(fit1.pred, countries.test.4$pov,
  k)
fit2.train.r2 <- pool.r.squared(fit2, adjusted = T)[,
  "est"]
fit2.test.r2 <- adjR2(fit2.pred, countries.test.4$pov,
  k)
res <- data.frame(model = c("Single", "Pooled"), train = c(fit1.train.r2,
  fit2.train.r2), test = c(fit1.test.r2, fit2.test.r2))
kable(res)

```

| | model | train | test |
|--------|-----------|-----------|------|
| Single | 0.7900486 | 0.5732851 | |
| Pooled | 0.7858952 | 0.5743992 | |

The models seem to be over-fitting. We should perform model simplification.

B. Regularization Prepare data

```

sd0 <- function(vct) {
  if (!is.numeric(vct)) {
    return(NA)
  }

  return(sd(vct, na.rm = T))
}

std0 <- function(vct, scl) {
  if (!is.numeric(vct)) {
    return(vct)
  }

  return(vct/scl)
}

standardise <- function(train, test) {
  data <- rbind(train, test)
  scaler <- unlist(lapply(data, sd0))

  numCols <- which(unlist(lapply(data, is.numeric)))
  num <- as.data.frame(mapply(std0, vct = data[, 

```

```

        numCols], scl = scaler[numCols], SIMPLIFY = T))
fct <- data[, -numCols]
return(cbind(fct, num))
}

optimalModel <- function(train) {
  # standardized
}

```

B. Step-wise AIC We can conduct a step-wise AIC variable selection. It is similar to the procedure we use in class, but based on a metric call AIC (Akaike Information Criterion), which is an estimator of prediction error and relative quality of statistical models. The lower AIC is, the better the model fits. \ We have no idea how to pool coefficients for AIC model,

```

# helper
fitAIC <- function(i) {
  fit <- lm(as.formula(formula.str), complete(countries1.imputed,
    i))
  aic.fit <- stepAIC(fit, trace = F, direction = "backward")
  return(aic.fit)
}

# build models
aic.fits <- lapply(1:countries1.imputed$m, fitAIC)
# predict with each model
aic.pred <- lapply(aic.fits, predict, newdata = countries.test.4)
# calculate adj r2
aic.train.r2 <- unlist(lapply(aic.fits, function(model) summary(model)$adj.r.squared))

k <- lapply(aic.fits, function(m) length(m$coefficients))
aic.test.r2 <- unlist(mapply(adjR2, aic.pred, k = k,
  MoreArgs = list(orig = countries.test.4$pov)))

res <- data.frame(`set no.` = 1:3, train = aic.train.r2,
  test = aic.test.r2)

```

Adjusted R-Squared

```
kable(res)
```

| set.no. | train | test |
|---------|-----------|-----------|
| 1 | 0.7901813 | 0.5745707 |
| 2 | 0.7865726 | 0.5630834 |
| 3 | 0.7815808 | 0.5853173 |

-> AIC on one model [link](#)

-> stepwise selection on mice [link](#)

3.2.4.2. Assessment Check assumption outliers ##### 3.2.4.3. Interpretation

3.3. Panel Data Analysis (To be done)

4. Conclusion

5. Appendix

6. References

- Akbar, Muhammad, Mukaram Khan, Haidar Farooqe, and Kaleemullah. 2019. “Public Spending, Education and Poverty: A Cross Country Analysis” 4 (April): 12–20.
- Alruhaymi, Abdullah Z, and Charles J Kim. 2021. “Why Can Multiple Imputations and How (MICE) Algorithm Work?” *Open J. Stat.* 11 (05): 759–77.
- Arel-Bundock, Vincent, and Krzysztof J. Pelc. 2018. “When Can Multiple Imputation Improve Regression Estimates?” *Political Analysis* 26 (2): 240–45. <https://doi.org/10.1017/pan.2017.43>.
- Gopalan, Bhuvaneswari. 2020. “Mice Algorithm to Impute Missing Values in a Dataset.” *Numpy Ninja*. Numpy Ninja. <https://www.numpyninja.com/post/mice-algorithm-to-impute-missing-values-in-a-dataset>.