1_pure_replication

Lucas Salamuni - 7429674

2025-08-21

Packages

```r
# I. Load required packages
packages <- c("dplyr", "knitr", "tinytex", "readxl", "tidyr", "fastDummies",
              "sandwich", "lmtest", "estimatr", "purrr", "tibble", "writexl",
              "readr", "stringr", "sf", "rnaturalearth", "dplyr", "units",
              "igraph", "countrycode", "geosphere", "haven", "glmnet",
              "gravity", "modelsummary", "sessioninfo")

# II. Install packages if not already installed
if(sum(as.numeric(!packages %in% installed.packages())) != 0){
  instalador <- packages[!packages %in% installed.packages()]
  for(i in 1:length(instalador)) {
    install.packages(instalador, dependencies = T)
    break()}
  sapply(packages, require, character = T)
} else {
  sapply(packages, require, character = T)
}
```

```
##         dplyr         knitr       tinytex        readxl         tidyr
##          TRUE          TRUE          TRUE          TRUE          TRUE
##   fastDummies      sandwich        lmtest      estimatr         purrr
##          TRUE          TRUE          TRUE          TRUE          TRUE
##        tibble       writexl         readr       stringr            sf
##          TRUE          TRUE          TRUE          TRUE          TRUE
## rnaturalearth         dplyr         units        igraph   countrycode
##          TRUE          TRUE          TRUE          TRUE          TRUE
##     geosphere         haven        glmnet       gravity  modelsummary
##          TRUE          TRUE          TRUE          TRUE          TRUE
##   sessioninfo
##          TRUE
```

Session info

```r
session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.4.1 (2024-06-14 ucrt)
##  os       Windows 11 x64 (build 26100)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  Portuguese_Brazil.utf8
##  ctype    Portuguese_Brazil.utf8
##  tz       Europe/Berlin
##  date     2025-08-22
##  pandoc   3.4 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package       * version  date (UTC) lib source
##  bdsmatrix       1.3-7    2024-03-02 [1] CRAN (R 4.4.0)
##  boot            1.3-30   2024-02-26 [2] CRAN (R 4.4.1)
##  cellranger      1.1.0    2016-07-27 [1] CRAN (R 4.4.1)
##  censReg         0.5-38   2024-05-20 [1] CRAN (R 4.4.3)
##  class           7.3-22   2023-05-03 [2] CRAN (R 4.4.1)
##  classInt        0.4-10   2023-09-05 [1] CRAN (R 4.4.1)
##  cli             3.6.3    2024-06-21 [1] CRAN (R 4.4.1)
##  coda            0.19-4.1 2024-01-31 [1] CRAN (R 4.4.1)
##  codetools       0.2-20   2024-03-31 [2] CRAN (R 4.4.1)
##  collapse        2.0.15   2024-07-08 [1] CRAN (R 4.4.1)
##  countrycode   * 1.6.1    2025-03-31 [1] CRAN (R 4.4.3)
##  data.table      1.15.4   2024-03-30 [1] CRAN (R 4.4.1)
##  DBI             1.2.3    2024-06-02 [1] CRAN (R 4.4.1)
##  digest          0.6.36   2024-06-23 [1] CRAN (R 4.4.1)
##  dplyr         * 1.1.4    2023-11-17 [1] CRAN (R 4.4.1)
##  e1071           1.7-16   2024-09-16 [1] CRAN (R 4.4.1)
##  emmeans         1.10.3   2024-07-01 [1] CRAN (R 4.4.1)
##  estimability    1.5.1    2024-05-12 [1] CRAN (R 4.4.1)
##  estimatr      * 1.0.4    2024-03-31 [1] CRAN (R 4.4.2)
##  evaluate        0.24.0   2024-06-10 [1] CRAN (R 4.4.1)
##  fansi           1.0.6    2023-12-08 [1] CRAN (R 4.4.1)
##  fastDummies   * 1.7.5    2025-01-20 [1] CRAN (R 4.4.3)
##  fastmap         1.2.0    2024-05-15 [1] CRAN (R 4.4.1)
##  forcats         1.0.0    2023-01-29 [1] CRAN (R 4.4.1)
##  foreach         1.5.2    2022-02-02 [1] CRAN (R 4.4.1)
##  Formula         1.2-5    2023-02-24 [1] CRAN (R 4.4.0)
##  generics        0.1.3    2022-07-05 [1] CRAN (R 4.4.1)
##  geosphere     * 1.5-20   2024-10-04 [1] CRAN (R 4.4.3)
##  glmmML          1.1.7    2024-09-20 [1] CRAN (R 4.4.3)
##  glmnet        * 4.1-8    2023-08-22 [1] CRAN (R 4.4.1)
##  glue            1.7.0    2024-01-09 [1] CRAN (R 4.4.1)
##  gravity       * 1.1      2023-05-02 [1] CRAN (R 4.4.3)
##  haven         * 2.5.4    2023-11-30 [1] CRAN (R 4.4.1)
##  hms             1.1.3    2023-03-21 [1] CRAN (R 4.4.1)
##  htmltools       0.5.8.1  2024-04-04 [1] CRAN (R 4.4.1)
##  httr            1.4.7    2023-08-15 [1] CRAN (R 4.4.1)
##  igraph        * 2.0.3    2024-03-13 [1] CRAN (R 4.4.1)
##  insight         1.0.0    2024-11-26 [1] CRAN (R 4.4.2)
##  iterators       1.0.14   2022-02-05 [1] CRAN (R 4.4.1)
##  KernSmooth      2.23-24  2024-05-17 [2] CRAN (R 4.4.1)
##  knitr         * 1.48     2024-07-07 [1] CRAN (R 4.4.1)
##  lattice         0.22-6   2024-03-20 [2] CRAN (R 4.4.1)
##  lifecycle       1.0.4    2023-11-07 [1] CRAN (R 4.4.1)
##  lmtest        * 0.9-40   2022-03-21 [1] CRAN (R 4.4.1)
##  magrittr        2.0.3    2022-03-30 [1] CRAN (R 4.4.1)
##  MASS            7.3-60.2 2024-04-26 [2] CRAN (R 4.4.1)
##  Matrix        * 1.7-0    2024-04-26 [2] CRAN (R 4.4.1)
##  maxLik          1.5-2.1  2024-03-24 [1] CRAN (R 4.4.1)
##  miscTools       0.6-28   2023-05-03 [1] CRAN (R 4.4.1)
##  modelsummary  * 2.2.0    2024-09-02 [1] CRAN (R 4.4.2)
##  multcomp        1.4-26   2024-07-18 [1] CRAN (R 4.4.1)
##  multiwayvcov    1.2.3    2016-05-05 [1] CRAN (R 4.4.1)
##  mvtnorm         1.2-5    2024-05-21 [1] CRAN (R 4.4.1)
##  nlme            3.1-164  2023-11-27 [2] CRAN (R 4.4.1)
##  pillar          1.9.0    2023-03-22 [1] CRAN (R 4.4.1)
##  pkgconfig       2.0.3    2019-09-22 [1] CRAN (R 4.4.1)
##  plm             2.6-4    2024-04-01 [1] CRAN (R 4.4.1)
##  proxy           0.4-27   2022-06-09 [1] CRAN (R 4.4.1)
##  purrr         * 1.0.2    2023-08-10 [1] CRAN (R 4.4.1)
##  R6              2.5.1    2021-08-19 [1] CRAN (R 4.4.1)
##  rbibutils       2.2.16   2023-10-25 [1] CRAN (R 4.4.1)
##  Rcpp            1.0.13   2024-07-17 [1] CRAN (R 4.4.1)
##  Rdpack          2.6      2023-11-08 [1] CRAN (R 4.4.1)
##  readr         * 2.1.5    2024-01-10 [1] CRAN (R 4.4.1)
##  readxl        * 1.4.3    2023-07-06 [1] CRAN (R 4.4.1)
##  rlang           1.1.4    2024-06-04 [1] CRAN (R 4.4.1)
##  rmarkdown       2.27     2024-05-17 [1] CRAN (R 4.4.1)
##  rnaturalearth * 1.1.0    2025-07-28 [1] CRAN (R 4.4.3)
##  rstudioapi      0.16.0   2024-03-24 [1] CRAN (R 4.4.1)
##  sandwich      * 3.1-0    2023-12-11 [1] CRAN (R 4.4.1)
##  sessioninfo   * 1.2.2    2021-12-06 [1] CRAN (R 4.4.2)
##  sf            * 1.0-17   2024-09-06 [1] CRAN (R 4.4.1)
##  shape           1.4.6.1  2024-02-23 [1] CRAN (R 4.4.0)
##  sp              2.1-4    2024-04-30 [1] CRAN (R 4.4.1)
##  stringi         1.8.4    2024-05-06 [1] CRAN (R 4.4.0)
##  stringr       * 1.5.1    2023-11-14 [1] CRAN (R 4.4.1)
##  survival        3.6-4    2024-04-24 [2] CRAN (R 4.4.1)
##  tables          0.9.31   2024-08-29 [1] CRAN (R 4.4.2)
##  texreg          1.39.4   2024-07-24 [1] CRAN (R 4.4.1)
##  TH.data         1.1-2    2023-04-17 [1] CRAN (R 4.4.1)
##  tibble        * 3.2.1    2023-03-20 [1] CRAN (R 4.4.1)
##  tidyr         * 1.3.1    2024-01-24 [1] CRAN (R 4.4.1)
##  tidyselect      1.2.1    2024-03-11 [1] CRAN (R 4.4.1)
##  tinytex       * 0.52     2024-07-18 [1] CRAN (R 4.4.1)
##  tzdb            0.4.0    2023-05-12 [1] CRAN (R 4.4.1)
##  units         * 0.8-5    2023-11-28 [1] CRAN (R 4.4.1)
##  utf8            1.2.4    2023-10-22 [1] CRAN (R 4.4.1)
##  vctrs           0.6.5    2023-12-01 [1] CRAN (R 4.4.1)
##  writexl       * 1.5.0    2024-02-09 [1] CRAN (R 4.4.1)
##  xfun            0.46     2024-07-18 [1] CRAN (R 4.4.1)
##  xtable          1.8-4    2019-04-21 [1] CRAN (R 4.4.1)
##  yaml            2.3.10   2024-07-26 [1] CRAN (R 4.4.1)
##  zoo           * 1.8-12   2023-04-13 [1] CRAN (R 4.4.1)
##
##  [1] C:/Users/Lucas/AppData/Local/R/win-library/4.4
##  [2] C:/Program Files/R/R-4.4.1/library
##
## ------------------------------------------------------------------------------
```

Part 1. Pure Replication

1.1. Retrieve original data

```r
# I. Load the original RData file
load("Datasets/final08_1.RData")

# II. Convert to tibble for easier manipulation
df <- x %>%
  as_tibble()

# III. Remove original object from memory
rm(x)

# IV. Sort data by country code and group
data <- df %>%
  arrange(contcod, group)

# V. Display summary statistics
summary(df)
```

```
##    contcod               year       year_survey       DummyY
##  Length:11737       Min.   :2008   Min.   :2004   Min.   :0.0000
##  Class :AsIs        1st Qu.:2008   1st Qu.:2008   1st Qu.:0.0000
##  Mode  :character   Median :2008   Median :2008   Median :1.0000
##                     Mean   :2008   Mean   :2008   Mean   :0.5144
##                     3rd Qu.:2008   3rd Qu.:2008   3rd Qu.:1.0000
##                     Max.   :2008   Max.   :2011   Max.   :1.0000
##
##      group           maxgroup           inc                lninc
##  Min.   :  1.00   Min.   : 54.00   Min.   :    16.72   Min.   : 2.817
##  1st Qu.: 25.00   1st Qu.:100.00   1st Qu.:   911.70   1st Qu.: 6.815
##  Median : 50.00   Median :100.00   Median :  2460.07   Median : 7.808
##  Mean   : 50.44   Mean   : 99.79   Mean   :  6316.11   Mean   : 7.864
##  3rd Qu.: 75.00   3rd Qu.:100.00   3rd Qu.:  7802.75   3rd Qu.: 8.962
##  Max.   :100.00   Max.   :100.00   Max.   :211296.72   Max.   :12.261
##
##       pop               gdpppp           lngdpppp           gini
##  Min.   : 0.00310   Min.   :  303.2   Min.   : 5.714   Min.   :0.2307
##  1st Qu.: 0.04526   1st Qu.: 2576.0   1st Qu.: 7.854   1st Qu.:0.3074
##  Median : 0.13479   Median : 7560.0   Median : 8.931   Median :0.3597
##  Mean   : 0.52334   Mean   :12886.0   Mean   : 8.823   Mean   :0.3785
##  3rd Qu.: 0.38534   3rd Qu.:18773.0   3rd Qu.: 9.840   3rd Qu.:0.4375
##  Max.   :13.25640   Max.   :73127.0   Max.   :11.200   Max.   :0.6721
##                     NA's   :200       NA's   :200
##       ayos
##  Min.   : 1.239
##  1st Qu.: 6.474
##  Median : 8.713
##  Mean   : 8.132
##  3rd Qu.:10.075
##  Max.   :12.749
##  NA's   :2654
```

```
# VI. Show first few rows
head(df, 10)
```

```
## # A tibble: 10 x 13
##    contcod   year year_survey DummyY group maxgroup   inc lninc    pop gdpppp
##    <I<chr>> <int>       <dbl>  <dbl> <int>    <dbl> <dbl> <dbl>  <dbl>  <dbl>
##  1 ALB       2008        2008      0     1      100  729.  6.59 0.0314   7297
##  2 ALB       2008        2008      0     2      100  917.  6.82 0.0314   7297
##  3 ALB       2008        2008      0     3      100 1011.  6.92 0.0314   7297
##  4 ALB       2008        2008      0     4      100 1087.  6.99 0.0314   7297
##  5 ALB       2008        2008      0     5      100 1133.  7.03 0.0314   7297
##  6 ALB       2008        2008      0     6      100 1171.  7.07 0.0314   7297
##  7 ALB       2008        2008      0     7      100 1201.  7.09 0.0314   7297
##  8 ALB       2008        2008      0     8      100 1241.  7.12 0.0314   7297
##  9 ALB       2008        2008      0     9      100 1286.  7.16 0.0314   7297
## 10 ALB       2008        2008      0    10      100 1325.  7.19 0.0314   7297
## # i 3 more variables: lngdpppp <dbl>, gini <dbl>, ayos <dbl>
```

1.2. Original STATA script

```
set logtype text
log using c:\branko\interyd\where\sent_to_Restat\for_release\Restat_results.txt, replace

/* Table 1: key results */
/* use final08.dta */

sort contcod group
tab  contcod, gen(Dcont)

regress lninc lngdpppp gini if  maxgroup==100
regress lninc ayos gini if  maxgroup==100
regress lninc Dcont1-Dcont117 gini if maxgroup==100


regress lninc lngdpppp gini if  maxgroup==100 [w=pop]
regress lninc Dcont1-Dcont117 gini if maxgroup==100 [w=pop]
regress lninc Dcont1-Dcont117 gini if maxgroup==100 [w=pop]
```

```
/* Table 4 and Annex Table 1: trade-off between income level and inequality, at different points of inc

/* creation of ventiles out of percentiles and running of regressions for "where are you?" */
/* use final08.dta */


sort contcod group

gen ventile=.
replace ventile=1 if group<6
replace ventile=2 if group>5 & group<11
replace ventile=3 if group>10 & group<16
replace ventile=4 if group>15 & group<21
replace ventile=5 if group>20 & group<26
replace ventile=6 if group>25 & group<31
replace ventile=7 if group>30 & group<36
replace ventile=8 if group>35 & group<41
replace ventile=9 if group>40 & group<46
replace ventile=10 if group>45 & group<51
replace ventile=11 if group>50 & group<56
replace ventile=12 if group>55 & group<61
replace ventile=13 if group>60 & group<66
replace ventile=14 if group>65 & group<71
replace ventile=15 if group>70 & group<76
replace ventile=16 if group>75 & group<81
replace ventile=17 if group>80 & group<86
replace ventile=18 if group>85 & group<91
replace ventile=19 if group>90 & group<95
replace ventile=20 if group>95

replace ventile=. if maxgroup~=100


by contcod: egen bb1=sum(inc) if ventile==1
by contcod: egen bb2=sum(inc) if ventile==2
by contcod: egen bb3=sum(inc) if ventile==3
by contcod: egen bb4=sum(inc) if ventile==4
by contcod: egen bb5=sum(inc) if ventile==5
by contcod: egen bb6=sum(inc) if ventile==6
by contcod: egen bb7=sum(inc) if ventile==7
by contcod: egen bb8=sum(inc) if ventile==8
by contcod: egen bb9=sum(inc) if ventile==9
by contcod: egen bb10=sum(inc) if ventile==10
by contcod: egen bb11=sum(inc) if ventile==11
by contcod: egen bb12=sum(inc) if ventile==12
by contcod: egen bb13=sum(inc) if ventile==13
by contcod: egen bb14=sum(inc) if ventile==14
by contcod: egen bb15=sum(inc) if ventile==15
by contcod: egen bb16=sum(inc) if ventile==16
by contcod: egen bb17=sum(inc) if ventile==17
by contcod: egen bb18=sum(inc) if ventile==18
by contcod: egen bb19=sum(inc) if ventile==19
by contcod: egen bb20=sum(inc) if ventile==20



for num 1/20: replace bbX=bbX/5

for num 1/20: gen lnbbX=ln(bbX)
gen gini2=gini*100

regress lnbb1 lngdpppp gini2 if group==1 & maxgroup==100, cluster(contcod)
regress lnbb2 lngdpppp gini2 if group==6 & maxgroup==100, cluster(contcod)
regress lnbb3 lngdpppp gini2 if group==11 & maxgroup==100, cluster(contcod)
regress lnbb4 lngdpppp gini2 if group==16 & maxgroup==100, cluster(contcod)
regress lnbb5 lngdpppp gini2 if group==21 & maxgroup==100, cluster(contcod)
regress lnbb6 lngdpppp gini2 if group==26 & maxgroup==100, cluster(contcod)
regress lnbb7 lngdpppp gini2 if group==31 & maxgroup==100, cluster(contcod)
regress lnbb8 lngdpppp gini2 if group==36 & maxgroup==100, cluster(contcod)
regress lnbb9 lngdpppp gini2 if group==41 & maxgroup==100, cluster(contcod)
regress lnbb10 lngdpppp gini2 if group==46 & maxgroup==100, cluster(contcod)
regress lnbb11 lngdpppp gini2 if group==51 & maxgroup==100, cluster(contcod)
regress lnbb12 lngdpppp gini2 if group==56 & maxgroup==100, cluster(contcod)
regress lnbb13 lngdpppp gini2 if group==61 & maxgroup==100, cluster(contcod)
regress lnbb14 lngdpppp gini2 if group==66 & maxgroup==100, cluster(contcod)
regress lnbb15 lngdpppp gini2 if group==71 & maxgroup==100, cluster(contcod)
regress lnbb16 lngdpppp gini2 if group==76 & maxgroup==100, cluster(contcod)
regress lnbb17 lngdpppp gini2 if group==81 & maxgroup==100, cluster(contcod)
regress lnbb18 lngdpppp gini2 if group==86 & maxgroup==100, cluster(contcod)
regress lnbb19 lngdpppp gini2 if group==91 & maxgroup==100, cluster(contcod)
regress lnbb20 lngdpppp gini2 if group==96 & maxgroup==100, cluster(contcod)


regress lnbb1 lngdpppp gini2 if group==1 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb2 lngdpppp gini2 if group==6 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb3 lngdpppp gini2 if group==11 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb4 lngdpppp gini2 if group==16 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb5 lngdpppp gini2 if group==21 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb6 lngdpppp gini2 if group==26 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb7 lngdpppp gini2 if group==31 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb8 lngdpppp gini2 if group==36 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb9 lngdpppp gini2 if group==41 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb10 lngdpppp gini2 if group==46 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb11 lngdpppp gini2 if group==51 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb12 lngdpppp gini2 if group==56 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb13 lngdpppp gini2 if group==61 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb14 lngdpppp gini2 if group==66 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb15 lngdpppp gini2 if group==71 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb16 lngdpppp gini2 if group==76 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb17 lngdpppp gini2 if group==81 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb18 lngdpppp gini2 if group==86 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb19 lngdpppp gini2 if group==91 & maxgroup==100 [w=pop], cluster(contcod)
regress lnbb20 lngdpppp gini2 if group==96 & maxgroup==100 [w=pop], cluster(contcod)
```

```
log close
```

1.3. Table 1: Key Results

```
# I. Create country dummy variables (equivalent to tab contcod, gen(Dcont))
data_with_dummies <- df %>%
  dummy_cols(select_columns = "contcod",
             remove_first_dummy = FALSE,
             remove_most_frequent_dummy = FALSE)

# II. Filter for complete cases (maxgroup == 100)
complete_data <- data_with_dummies %>%
  filter(maxgroup == 100)
```

1.4. Unweighted Regressions (Table 2)

```
# I. Regression 1: lninc ~ lngdpppp + gini
reg1 <- lm(data = complete_data,
              lninc ~ lngdpppp + gini)

# II. Display regression summary
summary(reg1)
```

```
##
## Call:
## lm(formula = lninc ~ lngdpppp + gini, data = complete_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9745 -0.4885  0.0072  0.4971  3.9768
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.799751   0.074135   10.79   <2e-16 ***
## lngdpppp     0.868168   0.006382  136.04   <2e-16 ***
## gini        -1.538326   0.088561  -17.37   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8136 on 11480 degrees of freedom
##   (200 observations deleted due to missingness)
## Multiple R-squared:  0.6602, Adjusted R-squared:  0.6602
## F-statistic: 1.115e+04 on 2 and 11480 DF,  p-value: < 2.2e-16
```

```
# III. Calculate clustered standard errors (by country)
coeftest(reg1, vcov = vcovCL(reg1, cluster = ~ contcod))
```

```
## 
## t test of coefficients:
## 
##              Estimate Std. Error t value  Pr(>|t|)    
## (Intercept)  0.799751   0.339104  2.3584   0.01837 *  
## lngdpppp     0.868168   0.029845 29.0896 < 2.2e-16 ***
## gini        -1.538326   0.353655 -4.3498 1.374e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```r
# IV. Regression 2: lninc ~ ayos + gini
reg2 <- lm(data = complete_data,
           lninc ~ ayos + gini)

# V. Display regression summary
summary(reg2)
```

```
## 
## Call:
## lm(formula = lninc ~ ayos + gini, data = complete_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -6.3247 -0.6639  0.0178  0.6863  4.1968 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  5.779062   0.067943   85.06   <2e-16 ***
## ayos         0.334633   0.004177   80.11   <2e-16 ***
## gini        -1.266877   0.122405  -10.35   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.9915 on 9080 degrees of freedom
##   (2600 observations deleted due to missingness)
## Multiple R-squared:  0.481,  Adjusted R-squared:  0.4809 
## F-statistic:  4207 on 2 and 9080 DF,  p-value: < 2.2e-16
```

```r
# VI. Calculate clustered standard errors (by country)
coeftest(reg2, vcov = vcovCL(reg2, cluster = ~ contcod))
```

```
## 
## t test of coefficients:
## 
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  5.779062   0.373730 15.4632   <2e-16 ***
## ayos         0.334633   0.022083 15.1537   <2e-16 ***
## gini        -1.266877   0.707289 -1.7912   0.0733 .  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```r
# VII. Regression 3: lninc ~ country dummies + gini
# Get all country dummy column names
country_dummies <- names(complete_data)[grepl("^contcod_", names(complete_data))]

# VIII. Create formula with all country dummies
formula_str <- paste("lninc ~ ", paste(country_dummies, collapse = " + "))

# IX. Run regression with country fixed effects
reg3 <- lm(data = complete_data,
           as.formula(formula_str))

# X. Calculate clustered standard errors (by country)
coeftest(reg3, vcov = vcovCL(reg3, cluster = ~ contcod))
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error    t value  Pr(>|t|)
## (Intercept) 5.2880e+00 5.2609e-14 1.0052e+14 < 2.2e-16 ***
## contcod_ALB 2.5637e+00 6.4885e-14 3.9512e+13 < 2.2e-16 ***
## contcod_ARG 3.0141e+00 7.0463e-14 4.2775e+13 < 2.2e-16 ***
## contcod_ARM 1.9939e+00 5.3073e-14 3.7570e+13 < 2.2e-16 ***
## contcod_AUT 4.3020e+00 6.2597e-14 6.8725e+13 < 2.2e-16 ***
## contcod_AZE 2.3159e+00 5.5996e-14 4.1358e+13 < 2.2e-16 ***
## contcod_BEL 4.1893e+00 5.8858e-14 7.1178e+13 < 2.2e-16 ***
## contcod_BFA 1.2718e+00 5.3996e-14 2.3554e+13 < 2.2e-16 ***
## contcod_BGD 1.4566e+00 5.8877e-14 2.4740e+13 < 2.2e-16 ***
## contcod_BGR 2.9872e+00 5.5978e-14 5.3363e+13 < 2.2e-16 ***
## contcod_BIH 3.2518e+00 6.0016e-14 5.4181e+13 < 2.2e-16 ***
## contcod_BLR 2.8696e+00 6.4102e-14 4.4766e+13 < 2.2e-16 ***
## contcod_BOL 2.0710e+00 5.3264e-14 3.8882e+13 < 2.2e-16 ***
## contcod_BRA 2.6426e+00 5.5964e-14 4.7220e+13 < 2.2e-16 ***
## contcod_BTN 1.7968e+00 5.3090e-14 3.3844e+13 < 2.2e-16 ***
## contcod_CAF 8.4276e-01 5.5817e-14 1.5099e+13 < 2.2e-16 ***
## contcod_CAN 4.5265e+00 5.4247e-14 8.3443e+13 < 2.2e-16 ***
## contcod_CHE 4.7475e+00 5.4284e-14 8.7456e+13 < 2.2e-16 ***
## contcod_CHL 3.0744e+00 5.3857e-14 5.7085e+13 < 2.2e-16 ***
## contcod_CHN 2.1100e+00 5.4555e-14 3.8676e+13 < 2.2e-16 ***
## contcod_CIV 4.0657e-01 5.5069e-14 7.3830e+12 < 2.2e-16 ***
## contcod_CMR 1.9576e+00 5.5571e-14 3.5227e+13 < 2.2e-16 ***
## contcod_COL 2.2684e+00 5.3769e-14 4.2188e+13 < 2.2e-16 ***
## contcod_CRI 2.9128e+00 5.4560e-14 5.3387e+13 < 2.2e-16 ***
## contcod_CYP 4.3419e+00 5.4658e-14 7.9437e+13 < 2.2e-16 ***
## contcod_CZE 3.6179e+00 5.2796e-14 6.8525e+13 < 2.2e-16 ***
## contcod_DEU 4.3522e+00 5.2806e-14 8.2419e+13 < 2.2e-16 ***
## contcod_DNK 4.3344e+00 5.3534e-14 8.0965e+13 < 2.2e-16 ***
## contcod_DOM 2.4454e+00 5.3361e-14 4.5828e+13 < 2.2e-16 ***
## contcod_ECU 2.3630e+00 5.2822e-14 4.4735e+13 < 2.2e-16 ***
## contcod_EGY 2.1164e+00 5.2831e-14 4.0059e+13 < 2.2e-16 ***
## contcod_ESP 4.0212e+00 5.3312e-14 7.5429e+13 < 2.2e-16 ***
## contcod_EST 3.5061e+00 5.2609e-14 6.6644e+13 < 2.2e-16 ***
## contcod_FIN 4.2822e+00 5.2923e-14 8.0914e+13 < 2.2e-16 ***
## contcod_FJI 2.0541e+00 5.2777e-14 3.8921e+13 < 2.2e-16 ***
## contcod_FRA 4.3441e+00 5.2735e-14 8.2375e+13 < 2.2e-16 ***
## contcod_GBR 4.4939e+00 5.2874e-14 8.4992e+13 < 2.2e-16 ***
## contcod_GEO 1.6597e+00 5.2612e-14 3.1546e+13 < 2.2e-16 ***
## contcod_GHA 9.9605e-01 5.2623e-14 1.8928e+13 < 2.2e-16 ***
## contcod_GIN 9.9441e-01 5.2638e-14 1.8891e+13 < 2.2e-16 ***
## contcod_GRC 3.8902e+00 5.2633e-14 7.3912e+13 < 2.2e-16 ***
## contcod_GTM 1.7793e+00 5.2643e-14 3.3800e+13 < 2.2e-16 ***
## contcod_HND 2.0930e+00 5.2627e-14 3.9770e+13 < 2.2e-16 ***
## contcod_HRV 3.5101e+00 5.2618e-14 6.6710e+13 < 2.2e-16 ***
## contcod_HUN 3.3011e+00 5.2620e-14 6.2735e+13 < 2.2e-16 ***
## contcod_IDN 1.6815e+00 5.2610e-14 3.1961e+13 < 2.2e-16 ***
## contcod_IND 1.3595e+00 5.2609e-14 2.5841e+13 < 2.2e-16 ***
## contcod_IRL 4.3392e+00 5.2686e-14 8.2360e+13 < 2.2e-16 ***
## contcod_IRN 3.0512e+00 5.2614e-14 5.7992e+13 < 2.2e-16 ***
## contcod_IRQ 1.9301e+00 5.2636e-14 3.6670e+13 < 2.2e-16 ***
## contcod_ISL 4.7735e+00 5.2615e-14 9.0725e+13 < 2.2e-16 ***
## contcod_ISR 3.6958e+00 5.2637e-14 7.0212e+13 < 2.2e-16 ***
## contcod_ITA 4.1431e+00 5.2613e-14 7.8747e+13 < 2.2e-16 ***
## contcod_JOR 2.5500e+00 5.2712e-14 4.8376e+13 < 2.2e-16 ***
## contcod_JPN 4.2925e+00 5.2727e-14 8.1410e+13 < 2.2e-16 ***
## contcod_KAZ 2.3103e+00 5.2887e-14 4.3683e+13 < 2.2e-16 ***
## contcod_KEN 8.1475e-01 5.2740e-14 1.5449e+13 < 2.2e-16 ***
## contcod_KGZ 1.9897e+00 5.2788e-14 3.7692e+13 < 2.2e-16 ***
## contcod_KHM 1.8565e+00 5.3033e-14 3.5007e+13 < 2.2e-16 ***
## contcod_KOR 4.0672e+00 5.2657e-14 7.7239e+13 < 2.2e-16 ***
## contcod_KOS 2.2349e+00 5.2657e-14 4.2442e+13 < 2.2e-16 ***
## contcod_LAO 1.4172e+00 5.3074e-14 2.6702e+13 < 2.2e-16 ***
## contcod_LBR 8.7843e-01 5.2609e-14 1.6697e+13 < 2.2e-16 ***
## contcod_LKA 1.9844e+00 5.2920e-14 3.7499e+13 < 2.2e-16 ***
## contcod_LTU 3.3141e+00 5.2667e-14 6.2926e+13 < 2.2e-16 ***
## contcod_LUX 4.7054e+00 5.3401e-14 8.8115e+13 < 2.2e-16 ***
## contcod_LVA 3.3030e+00 5.3028e-14 6.2289e+13 < 2.2e-16 ***
## contcod_MAR 2.1939e+00 5.3418e-14 4.1070e+13 < 2.2e-16 ***
## contcod_MDA 2.2243e+00 5.2641e-14 4.2254e+13 < 2.2e-16 ***
## contcod_MDG 2.2421e-01 5.3431e-14 4.1963e+12 < 2.2e-16 ***
## contcod_MEX 2.4932e+00 5.3739e-14 4.6394e+13 < 2.2e-16 ***
## contcod_MKD 2.8825e+00 5.2617e-14 5.4781e+13 < 2.2e-16 ***
## contcod_MLI 1.0587e+00 5.4207e-14 1.9531e+13 < 2.2e-16 ***
## contcod_MNE 3.3595e+00 5.2835e-14 6.3586e+13 < 2.2e-16 ***
## contcod_MNG 2.2592e+00 5.3374e-14 4.2329e+13 < 2.2e-16 ***
## contcod_MOZ 8.8835e-01 5.3456e-14 1.6618e+13 < 2.2e-16 ***
## contcod_MRT 1.9297e+00 5.2637e-14 3.6661e+13 < 2.2e-16 ***
## contcod_MWI 1.1870e+00 5.2612e-14 2.2561e+13 < 2.2e-16 ***
## contcod_MYS 3.0318e+00 5.2686e-14 5.7544e+13 < 2.2e-16 ***
## contcod_NER 9.6083e-01 5.4536e-14 1.7618e+13 < 2.2e-16 ***
## contcod_NGA 8.7018e-01 5.3451e-14 1.6280e+13 < 2.2e-16 ***
## contcod_NIC 2.0970e+00 5.4630e-14 3.8385e+13 < 2.2e-16 ***
## contcod_NLD 4.3534e+00 5.4813e-14 7.9423e+13 < 2.2e-16 ***
## contcod_NOR 4.6116e+00 5.2779e-14 8.7376e+13 < 2.2e-16 ***
## contcod_NPL 1.3555e+00 5.3295e-14 2.5434e+13 < 2.2e-16 ***
## contcod_PAK 1.3548e+00 5.3924e-14 2.5124e+13 < 2.2e-16 ***
## contcod_PAN 2.7344e+00 5.2641e-14 5.1944e+13 < 2.2e-16 ***
## contcod_PER 2.4077e+00 5.3764e-14 4.4782e+13 < 2.2e-16 ***
## contcod_PHL 1.6848e+00 5.2985e-14 3.1797e+13 < 2.2e-16 ***
## contcod_POL 3.1811e+00 5.2992e-14 6.0030e+13 < 2.2e-16 ***
## contcod_PRT 3.7093e+00 5.2659e-14 7.0440e+13 < 2.2e-16 ***
## contcod_PRY 2.3045e+00 5.2699e-14 4.3729e+13 < 2.2e-16 ***
## contcod_ROM 2.5617e+00 5.2882e-14 4.8441e+13 < 2.2e-16 ***
## contcod_RUS 3.2996e+00 5.3516e-14 6.1656e+13 < 2.2e-16 ***
## contcod_SDN 1.5009e+00 5.2889e-14 2.8378e+13 < 2.2e-16 ***
## contcod_SLV 2.2709e+00 5.2670e-14 4.3116e+13 < 2.2e-16 ***
## contcod_SRB 3.0805e+00 5.3086e-14 5.8028e+13 < 2.2e-16 ***
## contcod_SVK 3.3159e+00 5.2633e-14 6.3000e+13 < 2.2e-16 ***
## contcod_SVN 4.0242e+00 5.2935e-14 7.6021e+13 < 2.2e-16 ***
## contcod_SWE 4.2908e+00 5.3688e-14 7.9920e+13 < 2.2e-16 ***
## contcod_SWZ 6.1066e-01 5.3878e-14 1.1334e+13 < 2.2e-16 ***
## contcod_SYR 1.0349e+00 5.6820e-14 1.8214e+13 < 2.2e-16 ***
## contcod_THA 2.2210e+00 5.2621e-14 4.2208e+13 < 2.2e-16 ***
## contcod_TJK 2.1935e+00 5.2620e-14 4.1685e+13 < 2.2e-16 ***
## contcod_TLS 1.1360e+00 5.2689e-14 2.1560e+13 < 2.2e-16 ***
## contcod_TUR 3.1014e+00 5.2609e-14 5.8951e+13 < 2.2e-16 ***
## contcod_TWN 4.2621e+00 5.2689e-14 8.0891e+13 < 2.2e-16 ***
## contcod_TZA 8.5416e-01 5.2621e-14 1.6232e+13 < 2.2e-16 ***
## contcod_UGA 1.3029e+00 5.2964e-14 2.4599e+13 < 2.2e-16 ***
## contcod_UKR 2.7234e+00 5.2661e-14 5.1715e+13 < 2.2e-16 ***
## contcod_URY 2.9220e+00 5.2651e-14 5.5498e+13 < 2.2e-16 ***
## contcod_USA 4.5054e+00 5.2630e-14 8.5605e+13 < 2.2e-16 ***
## contcod_VEN 2.4374e+00 5.2609e-14 4.6331e+13 < 2.2e-16 ***
## contcod_VNM 1.7053e+00 5.2610e-14 3.2415e+13 < 2.2e-16 ***
## contcod_WBG 1.5213e+00 5.2609e-14 2.8918e+13 < 2.2e-16 ***
## contcod_YEM 1.4277e+00 5.2611e-14 2.7136e+13 < 2.2e-16 ***
## contcod_ZAF 2.4297e+00 5.2609e-14 4.6185e+13 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# XI. Display regression summary
summary(reg3)
```

```
##
## Call:
## lm(formula = as.formula(formula_str), data = complete_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5034 -0.4167 -0.0109  0.4114  3.6017
##
## Coefficients: (2 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.28800    0.07215  73.287  < 2e-16 ***
## contcod_ALB  2.56371    0.10204  25.124  < 2e-16 ***
## contcod_ARG  3.01405    0.10204  29.537  < 2e-16 ***
## contcod_ARM  1.99395    0.10204  19.540  < 2e-16 ***
## contcod_AUT  4.30196    0.10204  42.159  < 2e-16 ***
## contcod_AZE  2.31590    0.10204  22.696  < 2e-16 ***
## contcod_BEL  4.18934    0.10204  41.055  < 2e-16 ***
## contcod_BFA  1.27184    0.10204  12.464  < 2e-16 ***
## contcod_BGD  1.45660    0.10204  14.275  < 2e-16 ***
## contcod_BGR  2.98715    0.10204  29.274  < 2e-16 ***
## contcod_BIH  3.25177    0.10204  31.867  < 2e-16 ***
## contcod_BLR  2.86956    0.10204  28.121  < 2e-16 ***
## contcod_BOL  2.07103    0.10204  20.296  < 2e-16 ***
## contcod_BRA  2.64263    0.10204  25.897  < 2e-16 ***
## contcod_BTN  1.79677    0.10204  17.608  < 2e-16 ***
## contcod_CAF  0.84276    0.10204   8.259  < 2e-16 ***
## contcod_CAN  4.52650    0.10204  44.359  < 2e-16 ***
## contcod_CHE  4.74746    0.10679  44.456  < 2e-16 ***
## contcod_CHL  3.07442    0.10204  30.129  < 2e-16 ***
## contcod_CHN  2.10995    0.10204  20.677  < 2e-16 ***
## contcod_CIV  0.40657    0.10204   3.984 6.81e-05 ***
## contcod_CMR  1.95759    0.10204  19.184  < 2e-16 ***
## contcod_COL  2.26837    0.10204  22.230  < 2e-16 ***
## contcod_CRI  2.91280    0.10204  28.545  < 2e-16 ***
## contcod_CYP  4.34192    0.10204  42.550  < 2e-16 ***
## contcod_CZE  3.61788    0.10204  35.455  < 2e-16 ***
## contcod_DEU  4.35219    0.10204  42.651  < 2e-16 ***
## contcod_DNK  4.33436    0.10204  42.476  < 2e-16 ***
## contcod_DOM  2.44541    0.10204  23.965  < 2e-16 ***
## contcod_ECU  2.36296    0.10204  23.157  < 2e-16 ***
## contcod_EGY  2.11637    0.10204  20.740  < 2e-16 ***
## contcod_ESP  4.02123    0.10204  39.408  < 2e-16 ***
## contcod_EST  3.50609    0.10204  34.359  < 2e-16 ***
## contcod_FIN  4.28223    0.10204  41.965  < 2e-16 ***
## contcod_FJI  2.05415    0.10204  20.130  < 2e-16 ***
## contcod_FRA  4.34405    0.10204  42.571  < 2e-16 ***
## contcod_GBR  4.49387    0.10204  44.039  < 2e-16 ***
## contcod_GEO  1.65968    0.10204  16.265  < 2e-16 ***
## contcod_GHA  0.99605    0.10204   9.761  < 2e-16 ***
## contcod_GIN  0.99441    0.10204   9.745  < 2e-16 ***
## contcod_GRC  3.89021    0.10204  38.124  < 2e-16 ***
## contcod_GTM  1.77933    0.10204  17.437  < 2e-16 ***
## contcod_HND  2.09296    0.10204  20.511  < 2e-16 ***
## contcod_HRV  3.51012    0.10204  34.399  < 2e-16 ***
## contcod_HUN  3.30113    0.10204  32.351  < 2e-16 ***
## contcod_IDN  1.68148    0.10204  16.478  < 2e-16 ***
## contcod_IND  1.35948    0.10204  13.323  < 2e-16 ***
## contcod_IRL  4.33918    0.10204  42.524  < 2e-16 ***
## contcod_IRN  3.05122    0.10204  29.902  < 2e-16 ***
## contcod_IRQ  1.93013    0.10204  18.915  < 2e-16 ***
## contcod_ISL  4.77349    0.10204  46.780  < 2e-16 ***
## contcod_ISR  3.69578    0.10204  36.218  < 2e-16 ***
## contcod_ITA  4.14312    0.10204  40.602  < 2e-16 ***
## contcod_JOR  2.54999    0.10204  24.990  < 2e-16 ***
## contcod_JPN  4.29250    0.10204  42.066  < 2e-16 ***
## contcod_KAZ  2.31028    0.10204  22.641  < 2e-16 ***
## contcod_KEN  0.81475    0.10204   7.984 1.54e-15 ***
## contcod_KGZ  1.98968    0.10204  19.499  < 2e-16 ***
## contcod_KHM  1.85652    0.10204  18.194  < 2e-16 ***
## contcod_KOR  4.06717    0.10204  39.858  < 2e-16 ***
## contcod_KOS  2.23485    0.10204  21.901  < 2e-16 ***
## contcod_LAO  1.41720    0.10204  13.888  < 2e-16 ***
## contcod_LBR  0.87843    0.10204   8.609  < 2e-16 ***
## contcod_LKA  1.98442    0.10204  19.447  < 2e-16 ***
## contcod_LTU  3.31414    0.10230  32.396  < 2e-16 ***
## contcod_LUX  4.70540    0.10204  46.112  < 2e-16 ***
## contcod_LVA  3.30303    0.10204  32.369  < 2e-16 ***
## contcod_MAR  2.19388    0.10204  21.500  < 2e-16 ***
## contcod_MDA  2.22427    0.10204  21.798  < 2e-16 ***
## contcod_MDG  0.22421    0.10204   2.197    0.028 *  
## contcod_MEX  2.49317    0.10204  24.433  < 2e-16 ***
## contcod_MKD  2.88246    0.10204  28.248  < 2e-16 ***
## contcod_MLI  1.05870    0.10204  10.375  < 2e-16 ***
## contcod_MNE  3.35954    0.10204  32.923  < 2e-16 ***
## contcod_MNG  2.25925    0.10204  22.140  < 2e-16 ***
## contcod_MOZ  0.88835    0.10204   8.706  < 2e-16 ***
## contcod_MRT  1.92972    0.10204  18.911  < 2e-16 ***
## contcod_MWI  1.18700    0.10204  11.632  < 2e-16 ***
## contcod_MYS  3.03178    0.10204  29.711  < 2e-16 ***
## contcod_NER  0.96083    0.10204   9.416  < 2e-16 ***
## contcod_NGA  0.87018    0.10204   8.528  < 2e-16 ***
## contcod_NIC  2.09700    0.10204  20.550  < 2e-16 ***
## contcod_NLD  4.35341    0.10204  42.663  < 2e-16 ***
## contcod_NOR  4.61165    0.10204  45.194  < 2e-16 ***
## contcod_NPL  1.35550    0.10204  13.284  < 2e-16 ***
## contcod_PAK  1.35481    0.10204  13.277  < 2e-16 ***
## contcod_PAN  2.73436    0.10204  26.796  < 2e-16 ***
## contcod_PER  2.40766    0.10204  23.595  < 2e-16 ***
## contcod_PHL  1.68478    0.10204  16.511  < 2e-16 ***
## contcod_POL  3.18110    0.10204  31.174  < 2e-16 ***
## contcod_PRT  3.70927    0.10204  36.351  < 2e-16 ***
## contcod_PRY  2.30445    0.10204  22.583  < 2e-16 ***
## contcod_ROM  2.56166    0.10204  25.104  < 2e-16 ***
## contcod_RUS  3.29959    0.10204  32.336  < 2e-16 ***
## contcod_SDN  1.50088    0.10204  14.708  < 2e-16 ***
## contcod_SGP       NA         NA      NA       NA    
## contcod_SLV  2.27093    0.10204  22.255  < 2e-16 ***
## contcod_SRB  3.08047    0.10204  30.188  < 2e-16 ***
## contcod_SVK  3.31585    0.10204  32.495  < 2e-16 ***
## contcod_SVN  4.02423    0.10204  39.437  < 2e-16 ***
## contcod_SWE  4.29080    0.10204  42.049  < 2e-16 ***
## contcod_SWZ  0.61066    0.10204   5.984 2.24e-09 ***
## contcod_SYR  1.03490    0.10204  10.142  < 2e-16 ***
## contcod_THA  2.22101    0.10204  21.766  < 2e-16 ***
## contcod_TJK  2.19347    0.10204  21.496  < 2e-16 ***
## contcod_TLS  1.13596    0.10204  11.132  < 2e-16 ***
## contcod_TUR  3.10136    0.10204  30.393  < 2e-16 ***
## contcod_TWN  4.26209    0.10204  41.768  < 2e-16 ***
## contcod_TZA  0.85416    0.10204   8.371  < 2e-16 ***
## contcod_UGA  1.30287    0.10204  12.768  < 2e-16 ***
## contcod_UKR  2.72338    0.10204  26.689  < 2e-16 ***
## contcod_URY  2.92204    0.10204  28.636  < 2e-16 ***
## contcod_USA  4.50541    0.10204  44.153  < 2e-16 ***
## contcod_VEN  2.43741    0.10204  23.886  < 2e-16 ***
## contcod_VNM  1.70534    0.10204  16.712  < 2e-16 ***
## contcod_WBG  1.52133    0.10204  14.909  < 2e-16 ***
## contcod_YEM  1.42768    0.10204  13.991  < 2e-16 ***
## contcod_ZAF  2.42975    0.10204  23.811  < 2e-16 ***
## contcod_ZAR       NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.7215 on 11566 degrees of freedom
## Multiple R-squared:  0.7331, Adjusted R-squared:  0.7304
## F-statistic: 273.8 on 116 and 11566 DF,  p-value: < 2.2e-16
```

1.5. Creating Ventiles from Percentiles

```r
# I. Create ventiles (equivalent to Stata's ventile creation)
data_ventiles <- complete_data %>%
  mutate(ventile = case_when(
    group < 6 ~ 1,
    group >= 6 & group < 11 ~ 2,
    group >= 11 & group < 16 ~ 3,
    group >= 16 & group < 21 ~ 4,
    group >= 21 & group < 26 ~ 5,
    group >= 26 & group < 31 ~ 6,
    group >= 31 & group < 36 ~ 7,
    group >= 36 & group < 41 ~ 8,
    group >= 41 & group < 46 ~ 9,
    group >= 46 & group < 51 ~ 10,
    group >= 51 & group < 56 ~ 11,
    group >= 56 & group < 61 ~ 12,
    group >= 61 & group < 66 ~ 13,
    group >= 66 & group < 71 ~ 14,
    group >= 71 & group < 76 ~ 15,
    group >= 76 & group < 81 ~ 16,
    group >= 81 & group < 86 ~ 17,
    group >= 86 & group < 91 ~ 18,
    group >= 91 & group < 95 ~ 19,
    group >= 95 ~ 20,
    TRUE ~ NA_real_))

# II. Check ventile distribution
table(data_ventiles$ventile, useNA = "always")
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
##  580  580  582  585  585  585  584  585  584  584  585  584  585  585  585  585
##   17   18   19   20 <NA>
##  585  585  468  702    0
```

1.6. Calculate Average Income by Ventile within Country

```r
# I. Create all bb variables as NA
for(i in 1:20) {
  data_ventiles[[paste0("bb", i)]] <- NA_real_
}
```

```
# II. Calculate sum of income for each ventile within each country
for(i in 1:20) {
  ventile_sums <- data_ventiles %>%
    filter(ventile == i) %>%
    group_by(contcod) %>%
    summarise(sum_inc = sum(inc, na.rm = TRUE), .groups = "drop")

  # III. Assign the sum to all observations in that ventile for each country
  for(j in 1:nrow(ventile_sums)) {
    country <- ventile_sums$contcod[j]
    sum_val <- ventile_sums$sum_inc[j]

    data_ventiles[data_ventiles$contcod == country &
                  data_ventiles$ventile == i &
                  !is.na(data_ventiles$ventile), paste0("bb", i)] <- sum_val
  }
}

# IV. Divide by 5 to get average (equivalent to Stata's for num 1/20: replace bbX=bbX/5)
for(i in 1:20) {
  data_ventiles[[paste0("bb", i)]] <- data_ventiles[[paste0("bb", i)]] / 5
}

# V. Take log of average income (equivalent to Stata's for num 1/20: gen lnbbX=ln(bbX))
for(i in 1:20) {
  data_ventiles[[paste0("lnbb", i)]] <- log(data_ventiles[[paste0("bb", i)]])
}

# VI. Create gini2 (gini * 100)
data_ventiles <- data_ventiles %>%
  mutate(gini2 = gini * 100)

# Note: We no longer need separate ventile datasets since we're using the original data
# with the specific group filters in the regression functions
```

1.7. Ventile Regressions - Unweighted with Clustered Standard Errors

```
# I. Function to run regression for each ventile
run_ventile_regression <- function(ventile_num) {

  # II. Define specific group for each ventile
  ventile_groups <- c(1, 6, 11, 16,
                      21, 26, 31, 36,
                      41, 46, 51, 56,
                      61, 66, 71, 76,
                      81, 86, 91, 96)

  # III. Filter data for specific ventile group
  reg_data <- data_ventiles %>%
    filter(group == ventile_groups[ventile_num] & maxgroup == 100)
```

```r
  # IV. Run regression with clustered standard errors
  reg <- lm_robust(data = reg_data,
                   formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
                   clusters = contcod)

  return(reg)
}

# V. Run regressions for all 20 ventiles (unweighted)
ventile_regs_unweighted <- map(1:20, run_ventile_regression)

# VI. Name the regression results
names(ventile_regs_unweighted) <- paste0("ventile_", 1:20, "_reg")

# VII. Display results for first few ventiles
map(ventile_regs_unweighted[1:3], summary)
```

```
## $ventile_1_reg
##
## Call:
## lm_robust(formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
##     data = reg_data, clusters = contcod)
##
## Standard error type:  CR2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper    DF
## (Intercept)  1.87909   0.347926   5.401 2.257e-06  1.17878  2.57941 46.06
## lngdpppp     0.76863   0.030719  25.021 1.528e-30  0.70698  0.83028 51.66
## gini2       -0.05793   0.004085 -14.182 1.791e-15 -0.06624 -0.04961 32.50
##
## Multiple R-squared:  0.9011 ,    Adjusted R-squared:  0.8993
## F-statistic: 512.8 on 2 and 113 DF,  p-value: < 2.2e-16
##
## $ventile_2_reg
##
## Call:
## lm_robust(formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
##     data = reg_data, clusters = contcod)
##
## Standard error type:  CR2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper    DF
## (Intercept)  1.29432   0.342859   3.775 4.565e-04  0.60421  1.98444 46.06
## lngdpppp     0.82793   0.030727  26.945 4.351e-32  0.76626  0.88960 51.66
## gini2       -0.04433   0.003618 -12.251 1.005e-13 -0.05169 -0.03696 32.50
##
## Multiple R-squared:  0.9085 ,    Adjusted R-squared:  0.9069
## F-statistic: 545.1 on 2 and 113 DF,  p-value: < 2.2e-16
##
## $ventile_3_reg
##
```

```
## Call:
## lm_robust(formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
##     data = reg_data, clusters = contcod)
##
## Standard error type:  CR2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper    DF
## (Intercept)  1.09759   0.344570   3.185 2.594e-03  0.40403  1.79115 46.06
## lngdpppp     0.84544   0.030868  27.389 1.974e-32  0.78349  0.90739 51.66
## gini2       -0.03788   0.003538 -10.708 3.429e-12 -0.04508 -0.03068 32.50
##
## Multiple R-squared:  0.9087 ,    Adjusted R-squared:  0.907
## F-statistic:   540 on 2 and 113 DF,  p-value: < 2.2e-16
```

1.8. Ventile Regressions - Population-Weighted with Clustered Standard Errors

```
# I. Function to run weighted regression for each ventile
run_ventile_regression_weighted <- function(ventile_num) {

  # II. Define specific group for each ventile
  ventile_groups <- c(1, 6, 11, 16,
                      21, 26, 31, 36,
                      41, 46, 51, 56,
                      61, 66, 71, 76,
                      81, 86, 91, 96)

  # III. Filter data for specific ventile group
  reg_data <- data_ventiles %>%
    filter(group == ventile_groups[ventile_num] & maxgroup == 100)

  # IV. Run weighted regression with clustered standard errors
  reg <- lm_robust(data = reg_data,
                   formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
                   weights = pop,
                   clusters = contcod)

  return(reg)
}

# V. Run weighted regressions for all 20 ventiles
ventile_regs_weighted <- map(1:20, run_ventile_regression_weighted)

# VI. Name the regression results
names(ventile_regs_weighted) <- paste0("ventile_", 1:20, "_reg_weighted")

# VII. Display results for first few ventiles
map(ventile_regs_weighted[1:3], summary)
```

```
## $ventile_1_reg_weighted
##
## Call:
```

```
## lm_robust(formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
##     data = reg_data, weights = pop, clusters = contcod)
##
## Weighted, Standard error type:  CR2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper    DF
## (Intercept)  1.21615   0.576613   2.109 0.1659100 -1.20047  3.63278 2.057
## lngdpppp     0.84979   0.042996  19.765 0.0001456  0.72021  0.97937 3.324
## gini2       -0.06159   0.007525  -8.185 0.0113618 -0.09162 -0.03157 2.173
##
## Multiple R-squared:  0.9215 ,    Adjusted R-squared:  0.9201
## F-statistic: 322.1 on 2 and 113 DF,  p-value: < 2.2e-16
##
## $ventile_2_reg_weighted
##
## Call:
## lm_robust(formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
##     data = reg_data, weights = pop, clusters = contcod)
##
## Weighted, Standard error type:  CR2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper    DF
## (Intercept)  0.14840   0.717740  0.2068 0.8548702 -2.85970  3.15650 2.057
## lngdpppp     0.94935   0.052354 18.1331 0.0001935  0.79156  1.10714 3.324
## gini2       -0.04506   0.007697 -5.8540 0.0229697 -0.07577 -0.01435 2.173
##
## Multiple R-squared:  0.9313 ,    Adjusted R-squared:   0.93
## F-statistic: 329.9 on 2 and 113 DF,  p-value: < 2.2e-16
##
## $ventile_3_reg_weighted
##
## Call:
## lm_robust(formula = as.formula(paste0("lnbb", ventile_num, " ~ lngdpppp + gini2")),
##     data = reg_data, weights = pop, clusters = contcod)
##
## Weighted, Standard error type:  CR2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower  CI Upper    DF
## (Intercept) -0.18953   0.776456 -0.2441 0.8293578 -3.44372  3.064659 2.057
## lngdpppp     0.98075   0.056932 17.2268 0.0002291  0.80917  1.152334 3.324
## gini2       -0.03846   0.008208 -4.6861 0.0362520 -0.07121 -0.005714 2.173
##
## Multiple R-squared:  0.9311 ,    Adjusted R-squared:  0.9299
## F-statistic: 285.1 on 2 and 113 DF,  p-value: < 2.2e-16
```

1.9. Extract and Display Key Results (Table 4 + Table Appendix)

```
# I. Function to extract key statistics from regression results
extract_reg_stats <- function(reg_list, reg_names) {
  map_dfr(seq_along(reg_list), ~{
    reg <- reg_list[[.x]]
    tibble(regression = reg_names[.x],
           gdp_coef = reg$coefficients["lngdpppp"],
           gdp_se = reg$std.error["lngdpppp"],
           gdp_pvalue = reg$p.value["lngdpppp"],
           gini_coef = reg$coefficients["gini2"],
           gini_se = reg$std.error["gini2"],
           gini_pvalue = reg$p.value["gini2"],
           r_squared = reg$r.squared,
           adj_r_squared = reg$adj.r.squared,
           n_obs = reg$nobs)
  })
}

# II. Extract results for all ventile regressions
results_unweighted <- extract_reg_stats(ventile_regs_unweighted,
                                        paste0("Ventile ", 1:20, " (Unweighted)"))

results_weighted <- extract_reg_stats(ventile_regs_weighted,
                                      paste0("Ventile ", 1:20, " (Weighted)"))

# III. Display unweighted results
print(results_unweighted)
```

```
## # A tibble: 20 x 10
##    regression gdp_coef gdp_se gdp_pvalue gini_coef gini_se gini_pvalue r_squared
##    <chr>         <dbl>  <dbl>      <dbl>     <dbl>   <dbl>       <dbl>     <dbl>
##  1 Ventile 1~    0.769 0.0307   1.53e-30 -0.0579   0.00408    1.79e-15     0.901
##  2 Ventile 2~    0.828 0.0307   4.35e-32 -0.0443   0.00362    1.01e-13     0.909
##  3 Ventile 3~    0.845 0.0309   1.97e-32 -0.0379   0.00354    3.43e-12     0.909
##  4 Ventile 4~    0.857 0.0310   8.73e-33 -0.0336   0.00356    8.38e-11     0.908
##  5 Ventile 5~    0.865 0.0311   5.79e-33 -0.0299   0.00361    1.70e- 9     0.906
##  6 Ventile 6~    0.871 0.0309   3.22e-33 -0.0266   0.00365    2.69e- 8     0.905
##  7 Ventile 7~    0.873 0.0307   2.24e-33 -0.0237   0.00367    2.69e- 7     0.904
##  8 Ventile 8~    0.878 0.0309   2.12e-33 -0.0211   0.00369    2.37e- 6     0.902
##  9 Ventile 9~    0.881 0.0309   2.12e-33 -0.0186   0.00373    1.94e- 5     0.901
## 10 Ventile 1~    0.881 0.0306   1.24e-33 -0.0160   0.00371    1.37e- 4     0.900
## 11 Ventile 1~    0.885 0.0307   1.15e-33 -0.0135   0.00370    8.78e- 4     0.900
## 12 Ventile 1~    0.883 0.0304   7.82e-34 -0.0111   0.00368    5.01e- 3     0.899
## 13 Ventile 1~    0.886 0.0306   8.27e-34 -0.00872  0.00369    2.43e- 2     0.898
## 14 Ventile 1~    0.886 0.0304   6.20e-34 -0.00614  0.00367    1.04e- 1     0.897
## 15 Ventile 1~    0.886 0.0303   5.30e-34 -0.00341  0.00366    3.59e- 1     0.896
## 16 Ventile 1~    0.886 0.0302   4.36e-34 -0.000355 0.00363    9.23e- 1     0.895
## 17 Ventile 1~    0.885 0.0301   3.88e-34  0.00306  0.00364    4.07e- 1     0.894
## 18 Ventile 1~    0.882 0.0298   2.84e-34  0.00741  0.00365    5.02e- 2     0.893
## 19 Ventile 1~    0.876 0.0301   6.22e-34  0.0126   0.00364    1.50e- 3     0.889
## 20 Ventile 2~    0.863 0.0315   1.22e-32  0.0278   0.00375    1.85e- 8     0.874
## # i 2 more variables: adj_r_squared <dbl>, n_obs <int>
```

```
# IV. Display weighted results
print(results_weighted)
```

```
## # A tibble: 20 x 10
##    regression gdp_coef gdp_se gdp_pvalue gini_coef gini_se gini_pvalue r_squared
##    <chr>         <dbl>  <dbl>      <dbl>     <dbl>   <dbl>       <dbl>     <dbl>
##  1 Ventile 1~    0.850 0.0430   0.000146 -0.0616   0.00753      0.0114     0.922
##  2 Ventile 2~    0.949 0.0524   0.000193 -0.0451   0.00770      0.0230     0.931
##  3 Ventile 3~    0.981 0.0569   0.000229 -0.0385   0.00821      0.0363     0.931
##  4 Ventile 4~    1.00  0.0597   0.000261 -0.0338   0.00853      0.0506     0.930
##  5 Ventile 5~    1.01  0.0621   0.000284 -0.0299   0.00881      0.0685     0.929
##  6 Ventile 6~    1.02  0.0641   0.000305 -0.0263   0.00904      0.0916     0.928
##  7 Ventile 7~    1.03  0.0658   0.000326 -0.0230   0.00915      0.118      0.928
##  8 Ventile 8~    1.04  0.0669   0.000338 -0.0197   0.00925      0.157      0.927
##  9 Ventile 9~    1.04  0.0677   0.000344 -0.0163   0.00920      0.208      0.927
## 10 Ventile 1~    1.04  0.0686   0.000359 -0.0127   0.00921      0.292      0.927
## 11 Ventile 1~    1.04  0.0690   0.000364 -0.00908  0.00926      0.423      0.927
## 12 Ventile 1~    1.04  0.0695   0.000372 -0.00539  0.00937      0.620      0.927
## 13 Ventile 1~    1.04  0.0698   0.000378 -0.00215  0.00956      0.841      0.927
## 14 Ventile 1~    1.04  0.0696   0.000376  0.00121  0.00970      0.911      0.927
## 15 Ventile 1~    1.04  0.0693   0.000372  0.00451  0.00978      0.687      0.927
## 16 Ventile 1~    1.04  0.0684   0.000360  0.00758  0.00962      0.508      0.928
## 17 Ventile 1~    1.04  0.0669   0.000339  0.0107   0.00932      0.360      0.930
## 18 Ventile 1~    1.03  0.0647   0.000310  0.0141   0.00878      0.239      0.932
## 19 Ventile 1~    1.02  0.0622   0.000284  0.0180   0.00820      0.150      0.934
## 20 Ventile 2~    0.993 0.0574   0.000235  0.0247   0.00910      0.103      0.933
## # i 2 more variables: adj_r_squared <dbl>, n_obs <int>
```

1.10. Create Summary Table

```r
# I. Combine results for comparison
combined_results <- bind_rows(results_unweighted %>%
                                mutate(weighting = "Unweighted"),
                              results_weighted %>%
                                mutate(weighting = "Weighted")) %>%
  mutate(ventile = rep(1:20, 2)) %>%
  select(ventile, weighting, gdp_coef, gdp_se, gini_coef, gini_se, r_squared, n_obs)

# II. Display formatted table
formatted_table <- combined_results %>%
  mutate(gdp_result = paste0(round(gdp_coef, 3), " (", round(gdp_se, 3), ")"),
         gini_result = paste0(round(gini_coef, 4), " (", round(gini_se, 4), ")")) %>%
  select(ventile, weighting, gdp_result, gini_result, r_squared, n_obs) %>%
  arrange(ventile, weighting)

# III. Display the formatted table
print(formatted_table)
```

```
## # A tibble: 40 x 6
##    ventile weighting  gdp_result    gini_result      r_squared n_obs
##      <int> <chr>      <chr>         <chr>                <dbl> <int>
##  1       1 Unweighted 0.769 (0.031) -0.0579 (0.0041)     0.901   114
##  2       1 Weighted   0.85 (0.043)  -0.0616 (0.0075)     0.922   114
##  3       2 Unweighted 0.828 (0.031) -0.0443 (0.0036)     0.909   114
##  4       2 Weighted   0.949 (0.052) -0.0451 (0.0077)     0.931   114
##  5       3 Unweighted 0.845 (0.031) -0.0379 (0.0035)     0.909   114
##  6       3 Weighted   0.981 (0.057) -0.0385 (0.0082)     0.931   114
##  7       4 Unweighted 0.857 (0.031) -0.0336 (0.0036)     0.908   115
##  8       4 Weighted   1.001 (0.06)  -0.0338 (0.0085)     0.930   115
##  9       5 Unweighted 0.865 (0.031) -0.0299 (0.0036)     0.906   115
## 10       5 Weighted   1.014 (0.062) -0.0299 (0.0088)     0.929   115
## # i 30 more rows
```