3_income_importance_for_brazilian_migration

Lucas Salamuni - 7429674

2025-08-21

Packages

```
# 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", "fixest", "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
##        fixest   sessioninfo
##          TRUE          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)
##  dreamerr        1.4.0      2023-12-21 [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)
##  fixest        * 0.12.1     2024-06-13 [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)
##  numDeriv        2016.8-1.1 2019-06-06 [1] CRAN (R 4.4.0)
##  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)
##  stringmagic     1.1.2      2024-04-30 [1] CRAN (R 4.4.1)
##  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 3. Income Importance for Brazilian Migration

Dataset creation (BRA)

```r
# I. Load Brazilian expats data
BRA <- read_excel(path = "Datasets/BRA_expats.xlsx",
                  sheet = "Sheet1")

load("Auxiliary/country_mapping.RData")

# II. Join with country mapping
BRA <- BRA %>%
  left_join(country_mapping, by = "cont") %>%
  rename(reg = reg.x) %>%
  dplyr::select(c(contcod, cont, reg, pop)) %>%
  filter(contcod != is.na(contcod)) %>%

  # III. Add official language for each country
  mutate(official_language = case_when(
    contcod == "ALB" ~ "Albanian",
    contcod == "DZA" ~ "Standard Arabic",
    contcod == "AGO" ~ "Portuguese",
    contcod == "ARG" ~ "Spanish",
    contcod == "ARM" ~ "Armenian",
    contcod == "AUS" ~ "English",
    contcod == "AUT" ~ "German",
    contcod == "AZE" ~ "Azerbaijani",
    contcod == "BGD" ~ "Bengali",
    contcod == "BLR" ~ "Belarusian",
    contcod == "BEL" ~ "Dutch",
    contcod == "BEN" ~ "French",
    contcod == "BTN" ~ "Dzongkha",
    contcod == "BOL" ~ "Spanish",
    contcod == "BIH" ~ "Bosnian Standard",
    contcod == "BWA" ~ "English",
    contcod == "BRA" ~ "Portuguese",
    contcod == "BGR" ~ "Bulgarian",
    contcod == "BFA" ~ "French",
    contcod == "BDI" ~ "French",
    contcod == "KHM" ~ "Khmer",
    contcod == "CMR" ~ "French",
    contcod == "CAN" ~ "English",
    contcod == "CPV" ~ "Portuguese",
    contcod == "CAF" ~ "French",
    contcod == "TCD" ~ "French",
    contcod == "CHL" ~ "Spanish",
    contcod == "CHN" ~ "Chinese",
    contcod == "COL" ~ "Spanish",
    contcod == "COM" ~ "Standard Arabic",
    contcod == "COG" ~ "French",
    contcod == "CRI" ~ "Spanish",
    contcod == "CIV" ~ "French",
    contcod == "HRV" ~ "Croatian Standard",
    contcod == "CZE" ~ "Czech",
    contcod == "COD" ~ "French",
    contcod == "DNK" ~ "Danish",
    contcod == "DJI" ~ "French",
    contcod == "DOM" ~ "Spanish",
    contcod == "ECU" ~ "Spanish",
    contcod == "EGY" ~ "Standard Arabic",
    contcod == "SLV" ~ "Spanish",
    contcod == "EST" ~ "Estonian",
    contcod == "ETH" ~ "Amharic",
    contcod == "FJI" ~ "English",
    contcod == "FIN" ~ "Finnish",
    contcod == "FRA" ~ "French",
    contcod == "GAB" ~ "French",
    contcod == "GMB" ~ "English",
    contcod == "GEO" ~ "Georgian",
    contcod == "DEU" ~ "German",
    contcod == "GHA" ~ "English",
    contcod == "GRC" ~ "Greek",
    contcod == "GTM" ~ "Spanish",
    contcod == "GIN" ~ "French",
    contcod == "GNB" ~ "Portuguese",
    contcod == "GUY" ~ "English",
    contcod == "HTI" ~ "French",
    contcod == "HND" ~ "Spanish",
    contcod == "HKG" ~ "Chinese",
    contcod == "HUN" ~ "Hungarian",
    contcod == "ISL" ~ "Icelandic",
    contcod == "IND" ~ "Hindi",
    contcod == "IDN" ~ "Standard Indonesian",
    contcod == "IRN" ~ "Persian",
    contcod == "IRQ" ~ "Standard Arabic",
    contcod == "IRL" ~ "English",
    contcod == "ISR" ~ "Hebrew",
    contcod == "ITA" ~ "Italian",
    contcod == "JAM" ~ "English",
    contcod == "JPN" ~ "Japanese",
    contcod == "JOR" ~ "Standard Arabic",
    contcod == "KAZ" ~ "Kazakh",
    contcod == "KEN" ~ "Swahili",
    contcod == "KGZ" ~ "Kirghiz",
    contcod == "LAO" ~ "Lao",
    contcod == "LVA" ~ "Standard Latvian",
    contcod == "LBN" ~ "Standard Arabic",
    contcod == "LSO" ~ "English",
    contcod == "LBR" ~ "English",
    contcod == "LTU" ~ "Lithuanian",
    contcod == "LUX" ~ "French",
    contcod == "MKD" ~ "Macedonian",
    contcod == "MDG" ~ "Malagasy",
    contcod == "MWI" ~ "English",
    contcod == "MYS" ~ "Standard Malay",
    contcod == "MDV" ~ "Dhivehi",
    contcod == "MLI" ~ "French",
    contcod == "MLT" ~ "Maltese",
    contcod == "MRT" ~ "Standard Arabic",
    contcod == "MUS" ~ "English",
    contcod == "MEX" ~ "Spanish",
    contcod == "MDA" ~ "Romanian",
    contcod == "MNG" ~ "Mongolian",
    contcod == "MNE" ~ "Serbian",
    contcod == "MAR" ~ "Standard Arabic",
    contcod == "MOZ" ~ "Portuguese",
    contcod == "MMR" ~ "Burmese",
    contcod == "NAM" ~ "English",
    contcod == "NPL" ~ "Nepali",
    contcod == "NLD" ~ "Dutch",
    contcod == "NIC" ~ "Spanish",
    contcod == "NER" ~ "French",
    contcod == "NGA" ~ "English",
    contcod == "NOR" ~ "Norwegian",
    contcod == "PAK" ~ "Urdu",
    contcod == "PSE" ~ "Standard Arabic",
    contcod == "PAN" ~ "Spanish",
    contcod == "PNG" ~ "English",
    contcod == "PRY" ~ "Spanish",
    contcod == "PER" ~ "Spanish",
    contcod == "PHL" ~ "Filipino",
    contcod == "POL" ~ "Polish",
    contcod == "PRT" ~ "Portuguese",
    contcod == "ROM" ~ "Romanian",
    contcod == "RUS" ~ "Russian",
    contcod == "RWA" ~ "Kinyarwanda",
    contcod == "STP" ~ "Portuguese",
    contcod == "SEN" ~ "French",
    contcod == "SRB" ~ "Serbian",
    contcod == "SLE" ~ "English",
    contcod == "SGP" ~ "English",
    contcod == "SVK" ~ "Slovak",
    contcod == "SVN" ~ "Slovene",
    contcod == "ZAF" ~ "English",
    contcod == "KOR" ~ "Korean",
    contcod == "ESP" ~ "Spanish",
    contcod == "LKA" ~ "Sinhala",
    contcod == "SDN" ~ "Standard Arabic",
    contcod == "SWZ" ~ "English",
    contcod == "SWE" ~ "Swedish",
    contcod == "CHE" ~ "German",
    contcod == "SYR" ~ "Standard Arabic",
    contcod == "TWN" ~ "Chinese",
    contcod == "TJK" ~ "Tajik",
    contcod == "TZA" ~ "Swahili",
    contcod == "THA" ~ "Thai",
    contcod == "TLS" ~ "Portuguese",
    contcod == "TGO" ~ "French",
    contcod == "TTO" ~ "English",
    contcod == "TUN" ~ "Standard Arabic",
    contcod == "TUR" ~ "Turkish",
    contcod == "TKM" ~ "Turkmen",
    contcod == "UGA" ~ "English",
    contcod == "UKR" ~ "Ukrainian",
    contcod == "GBR" ~ "English",
    contcod == "USA" ~ "English",
    contcod == "URY" ~ "Spanish",
    contcod == "UZB" ~ "Uzbek",
    contcod == "VEN" ~ "Spanish",
    contcod == "VNM" ~ "Vietnamese",
    contcod == "YEM" ~ "Standard Arabic",
    contcod == "ZMB" ~ "English",
    contcod == "ZWE" ~ "English",
    contcod == "ARE" ~ "Standard Arabic",
    contcod == "AFG" ~ "Dari",
    contcod == "ATG" ~ "English",
    contcod == "AND" ~ "Catalan",
    contcod == "BHS" ~ "English",
    contcod == "BHR" ~ "Standard Arabic",
    contcod == "BRB" ~ "English",
    contcod == "BLZ" ~ "English",
    contcod == "BMU" ~ "English",
    contcod == "BRN" ~ "Standard Malay",
    contcod == "CYP" ~ "Greek",
    contcod == "DMA" ~ "English",
    contcod == "ERI" ~ "Tigrinya",
    contcod == "GRD" ~ "English",
    contcod == "GNQ" ~ "Spanish",
    contcod == "ISM" ~ "English",
    contcod == "KWT" ~ "Standard Arabic",
    contcod == "LIE" ~ "German",
    contcod == "MAC" ~ "Chinese",
    contcod == "MCO" ~ "French",
    contcod == "OMN" ~ "Standard Arabic",
    contcod == "PLW" ~ "Palauan",
    contcod == "QAT" ~ "Standard Arabic",
    contcod == "KNA" ~ "English",
    contcod == "LCA" ~ "English",
    contcod == "VCT" ~ "English",
    contcod == "WSM" ~ "Samoan",
    contcod == "SMR" ~ "Italian",
    contcod == "SAU" ~ "Standard Arabic",
    contcod == "SYC" ~ "English",
    contcod == "SOM" ~ "Somali",
    contcod == "SSD" ~ "English",
    contcod == "TON" ~ "Tongan",
    contcod == "VUT" ~ "Bislama",
    contcod == "VAT" ~ "Italian",
    contcod == "KOS" ~ "Albanian",
    TRUE ~ "Other"))
```

3.2. Linguistic Distance Index from PT-BR

```
# I. Load ASJP linguistic data
langs <- read_csv("Auxiliary/languages.csv")
```

```
## Rows: 11540 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (13): ID, Name, Glottocode, Glottolog_Name, ISO639P3code, Macroarea, Fam...
## dbl  (3): Latitude, Longitude, year_of_extinction
## lgl  (2): recently_extinct, long_extinct
##
## 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.
```

```
forms <- read_csv("Auxiliary/forms.csv")
```

```
## Rows: 568820 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): ID, Language_ID, Value, Form, Segments, Comment, Source, Graphemes...
## dbl  (1): Parameter_ID
## lgl  (3): Local_ID, Cognacy, Loan
##
## 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.
```

```
params <- read_csv("Auxiliary/parameters.csv")  # the 40 ASJP concepts
```

```
## Rows: 100 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Name, Concepticon_Gloss
## dbl (2): ID, Concepticon_ID
##
## 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.
```

```r
# II. Find Portuguese (using correct glottocode)
ptbr_id <- langs %>%
  filter(str_detect(Glottocode, "port1283")) %>% pull(ID)

if (length(ptbr_id) == 0) {
  ptbr_id <- langs %>%
    filter(str_detect(Name, "Portuguese")) %>% slice(1) %>% pull(ID)
}

# III. Helper function: get wordlist for one language ID
get_list <- function(lang_id) {
  forms %>%
    filter(Language_ID == lang_id) %>%
    inner_join(params, by = c("Parameter_ID" = "ID")) %>%
    group_by(Parameter_ID) %>%
    slice(1) %>%  # one form per concept
    ungroup() %>%
    select(Parameter_ID, Concept = Name, Word = Form)
}

# IV. Get Portuguese wordlist
pt <- get_list(ptbr_id)

# V. Normalized Levenshtein distance function
ldn <- function(x, y) {
  mean(mapply(function(a,b){
    if (is.na(a) || is.na(b)) return(NA_real_)
    adist(a,b) / max(nchar(a), nchar(b))
  }, x, y), na.rm = TRUE)
}

# VI. Create language name mapping from our names to ASJP database IDs
create_language_mapping <- function() {
  mapping <- tribble(
    ~our_name, ~asjp_id,
    "Standard Arabic", "STANDARD_ARABIC",
    "Portuguese", "PORTUGUESE",
    "French", "FRENCH",
    "English", "ENGLISH",
    "Spanish", "SPANISH",
    "German", "STANDARD_GERMAN",
    "Italian", "ITALIAN",
    "Russian", "RUSSIAN",
    "Japanese", "JAPANESE",
    "Chinese", "CANTONESE",  # Using Cantonese as fallback for Chinese
    "Hindi", "HINDI",
    "Bengali", "BENGALI",
    "Standard Indonesian", "INDONESIAN",
    "Standard Malay", "MELAYU",
    "Korean", "KOREAN",
    "Persian", "FARSI_WESTERN",
    "Hebrew", "HEBREW_MODERN",
    "Turkish", "TURKISH",
    "Dutch", "DUTCH",
    "Polish", "POLISH",
    "Czech", "CZECH",
    "Romanian", "ROMANIAN",
    "Bulgarian", "BULGARIAN",
    "Croatian Standard", "CROATIAN",
    "Bosnian Standard", "BOSNIAN",
    "Serbian", "CROATIAN",  # Using Croatian as fallback for Serbian
    "Hungarian", "HUNGARIAN",
    "Finnish", "FINNISH",
    "Swedish", "SWEDISH",
    "Norwegian", "NORWEGIAN_BOKMAAL",
    "Danish", "DANISH",
    "Greek", "GREEK",
    "Albanian", "ALBANIAN",
    "Estonian", "ESTONIAN",
    "Standard Latvian", "LATVIAN",
    "Lithuanian", "LITHUANIAN",
    "Ukrainian", "UKRAINIAN",
    "Slovak", "SLOVAK",
    "Slovene", "SLOVENIAN",
    "Icelandic", "ICELANDIC",
    "Vietnamese", "VIETNAMESE",
    "Thai", "THAI",
    "Catalan", "CATALAN",
    "Amharic", "AMHARIC",
    "Swahili", "SWAHILI",
    "Malagasy", "SAKALAVA",
    "Kinyarwanda", "KINYARWANDA",
    "Somali", "SOMALI_2",
    "Dari", "DARI",
    "Azerbaijani", "AZERBAIJANI_NORTH",
    "Khmer", "KHMER",
    "Kazakh", "KAZAKH",
    "Kirghiz", "KYRGYZ",
    "Dhivehi", "DHIVEHI",
    "Mongolian", "HALH",
    "Lao", "LAO",
    "Nepali", "NEPALI",
    "Sinhala", "SINHALA",
    "Urdu", "URDU",
    "Tajik", "TAJIKI",
    "Uzbek", "UZBEK",
    "Turkmen", "TURKMEN",
    "Armenian", "ARMENIAN_EASTERN",
    "Georgian", "GEORGIAN",
    "Belarusian", "BELARUSIAN",
    "Macedonian", "MACEDONIAN",
    "Montenegrin", "MONTENEGRIN",
```

```r
    "Luxembourgish", "LUXEMBURGEOIS",
    "Maltese", "MALTESE",
    "Malay", "MELAYU",
    "Burmese", "BURMESE",
    "Filipino", "TAGALOG",
    "Dzongkha", "DZONGKHA",
    "Tigrinya", "TIGRINYA",
    "Palauan", "PALAUAN",
    "Samoan", "SAMOAN",
    "Tongan", "TONGAN",
    "Bislama", "BISLAMA")
  return(mapping)
}

# VII. Function to calculate distance from PT-BR to target language
score_vs <- function(target_name_regex) {

  # Get language mapping
  lang_mapping <- create_language_mapping()

  # Find ASJP ID for target language
  asjp_id <- lang_mapping %>%
    filter(our_name == target_name_regex) %>%
    pull(asjp_id)

  if (length(asjp_id) == 0) {
    print(paste("Language not in mapping:", target_name_regex))
    return(tibble(target = target_name_regex, D = NA_real_, similarity = NA_real_, PPI = NA_real_))
  }

  # Find language ID in ASJP database
  tgt_id <- langs %>% filter(ID == asjp_id) %>% slice(1) %>% pull(ID)

  # If not found, try case-insensitive partial match on Name column
  if (length(tgt_id) == 0) {
    # Extract base language name for fuzzy matching
    base_name <- gsub("_.*", "", asjp_id)
    tgt_id <- langs %>%
      filter(str_detect(toupper(Name), toupper(base_name))) %>%
      slice(1) %>%
      pull(ID)
  }

  if (length(tgt_id) == 0) {
    print(paste("ASJP language not found:", asjp_id, "for", target_name_regex))
    return(tibble(target = target_name_regex, D = NA_real_, similarity = NA_real_, PPI = NA_real_))
  }

  # Get wordlist and calculate distance
  tgt <- get_list(tgt_id)
  both <- inner_join(pt, tgt, by = "Parameter_ID", suffix = c("_pt","_x"))

  if (nrow(both) == 0) {
    print(paste("No word overlap for:", target_name_regex))
    return(tibble(target = target_name_regex, D = NA_real_, similarity = NA_real_, PPI = NA_real_))
  }

  D <- ldn(both$Word_pt, both$Word_x)
  print(paste("Calculated distance for", target_name_regex, ":", round(100*(1-D), 2)))
  tibble(target = target_name_regex, D = D, similarity = 1 - D, PPI = 100*(1-D))
}

# VIII. Get unique languages from BRA dataset
BRA_languages <- unique(BRA$official_language)
BRA_languages <- BRA_languages[!is.na(BRA_languages) & BRA_languages != "Other"]

print(BRA_languages)
```

```
##  [1] "Standard Arabic"     "Portuguese"          "French"
##  [4] "English"             "Spanish"             "Amharic"
##  [7] "Swahili"             "Malagasy"            "Kinyarwanda"
## [10] "Somali"              "Dari"                "Azerbaijani"
## [13] "Bengali"             "Standard Malay"      "Khmer"
## [16] "Chinese"             "Hindi"               "Standard Indonesian"
## [19] "Persian"             "Hebrew"              "Japanese"
## [22] "Kazakh"              "Kirghiz"             "Dhivehi"
## [25] "Mongolian"           "Burmese"             "Nepali"
## [28] "Urdu"                "Filipino"            "Korean"
## [31] "Sinhala"             "Tajik"               "Thai"
## [34] "Turkish"             "Uzbek"               "Vietnamese"
## [37] "Albanian"            "Armenian"            "German"
## [40] "Belarusian"          "Dutch"               "Bosnian Standard"
## [43] "Bulgarian"           "Croatian Standard"   "Greek"
## [46] "Czech"               "Danish"              "Estonian"
## [49] "Finnish"             "Georgian"            "Hungarian"
## [52] "Icelandic"           "Italian"             "Standard Latvian"
## [55] "Lithuanian"          "Maltese"             "Romanian"
## [58] "Serbian"             "Macedonian"          "Norwegian"
## [61] "Polish"              "Russian"             "Slovak"
## [64] "Slovene"             "Swedish"             "Ukrainian"
## [67] "Bislama"
```

```
# IX. Display sample of available languages in ASJP
print(head(unique(langs$Name), 20))
```

```
##  [1] "A51_BAFIA_MAJA"        "A51_BAFIA_TUMI_TINGON" "A51_BAFIA_ZAKAAN"
##  [4] "A53_BAFIA_RIKPA"       "A54_BAFIA_NJANTI"      "A60_GUNU"
##  [7] "A60_MMAALA"            "A61_NGORO_ASOM"        "A62_KALONGE"
## [10] "A72a_EWONDO"           "AASAX"                 "ABAGA"
## [13] "ABANYOM"               "ABAR"                  "ABASAKUR"
## [16] "ABASOLO_VALLE_MIXTEC"  "ABAU"                  "ABAU_2"
## [19] "ABAWIRI"               "ABAZA"
```

```
# X. Calculate distances for all languages in BRA dataset
language_distances <- map_dfr(BRA_languages, score_vs) %>%
  rename(official_language = target)
```

```
## [1] "Calculated distance for Standard Arabic : 5.12"
## [1] "Calculated distance for Portuguese : 91.06"
## [1] "Calculated distance for French : 25.28"
## [1] "Calculated distance for English : 9.27"
## [1] "Calculated distance for Spanish : 38.31"
## [1] "Calculated distance for Amharic : 6.5"
## [1] "Calculated distance for Swahili : 8.28"
## [1] "Calculated distance for Malagasy : 12.14"
## [1] "Calculated distance for Kinyarwanda : 7.99"
## [1] "Calculated distance for Somali : 5.98"
## [1] "Calculated distance for Dari : 14.34"
## [1] "Calculated distance for Azerbaijani : 7.63"
## [1] "Calculated distance for Bengali : 18.1"
## [1] "Calculated distance for Standard Malay : 11.23"
## [1] "Calculated distance for Khmer : 7.7"
## [1] "Calculated distance for Chinese : 5.97"
## [1] "Calculated distance for Hindi : 16.41"
## [1] "Calculated distance for Standard Indonesian : 10.85"
## [1] "Calculated distance for Persian : 14.09"
## [1] "Calculated distance for Hebrew : 8.31"
## [1] "Calculated distance for Japanese : 6.2"
## [1] "Calculated distance for Kazakh : 8.99"
## [1] "Calculated distance for Kirghiz : 8.81"
## [1] "ASJP language not found: DHIVEHI for Dhivehi"
## [1] "ASJP language not found: HALH for Mongolian"
## [1] "Calculated distance for Burmese : 5.97"
## [1] "Calculated distance for Nepali : 14.36"
## [1] "Calculated distance for Urdu : 16.18"
## [1] "Calculated distance for Filipino : 8.55"
## [1] "Calculated distance for Korean : 6.2"
## [1] "Calculated distance for Sinhala : 16.53"
## [1] "Calculated distance for Tajik : 6.79"
## [1] "Calculated distance for Thai : 7.53"
## [1] "Calculated distance for Turkish : 11.76"
## [1] "Calculated distance for Uzbek : 5.7"
## [1] "Calculated distance for Vietnamese : 9"
## [1] "Calculated distance for Albanian : 17.14"
## [1] "Calculated distance for Armenian : 8.41"
## [1] "Calculated distance for German : 12.33"
## [1] "Calculated distance for Belarusian : 13.47"
## [1] "Calculated distance for Dutch : 10.98"
## [1] "Calculated distance for Bosnian Standard : 13.77"
## [1] "Calculated distance for Bulgarian : 14.14"
## [1] "Calculated distance for Croatian Standard : 13.05"
## [1] "Calculated distance for Greek : 14.64"
## [1] "Calculated distance for Czech : 13.28"
## [1] "Calculated distance for Danish : 9.35"
## [1] "Calculated distance for Estonian : 9.89"
## [1] "Calculated distance for Finnish : 9.49"
## [1] "Calculated distance for Georgian : 6.44"
## [1] "Calculated distance for Hungarian : 10.62"
## [1] "Calculated distance for Icelandic : 9.41"
## [1] "Calculated distance for Italian : 37.94"
## [1] "Calculated distance for Standard Latvian : 13.92"
## [1] "Calculated distance for Lithuanian : 12.39"
## [1] "Calculated distance for Maltese : 10.24"
## [1] "Calculated distance for Romanian : 33.44"
## [1] "Calculated distance for Serbian : 13.05"
## [1] "Calculated distance for Macedonian : 12.67"
## [1] "Calculated distance for Norwegian : 8.15"
## [1] "Calculated distance for Polish : 13.69"
## [1] "Calculated distance for Russian : 11.93"
## [1] "Calculated distance for Slovak : 16.21"
## [1] "Calculated distance for Slovene : 14.97"
## [1] "Calculated distance for Swedish : 8.47"
## [1] "Calculated distance for Ukrainian : 11.49"
## [1] "Calculated distance for Bislama : 13.71"
```

```
print(language_distances)
```

```
## # A tibble: 67 x 4
##    official_language      D similarity   PPI
##    <chr>              <dbl>      <dbl> <dbl>
##  1 Standard Arabic   0.949      0.0512  5.12
##  2 Portuguese        0.0894     0.911  91.1
##  3 French            0.747      0.253  25.3
##  4 English           0.907      0.0927  9.27
##  5 Spanish           0.617      0.383  38.3
##  6 Amharic           0.935      0.0650  6.50
##  7 Swahili           0.917      0.0828  8.28
##  8 Malagasy          0.879      0.121  12.1
##  9 Kinyarwanda       0.920      0.0799  7.99
## 10 Somali            0.940      0.0598  5.98
## # i 57 more rows
```

```
# XI. Add linguistic distance to BRA dataset
BRA <- BRA %>%
  left_join(language_distances, by = "official_language") %>%
  dplyr::select(-c(similarity, PPI, official_language)) %>%
  filter(D != is.na(D)) %>%
  rename(language_distance = D,
         expats = pop)

# XII. Remove redundant data
rm(forms, langs, language_distances, params, pt)
```

3.3. Cultural Proximity Index (based on foreign migration waves to Brazil between 1884 and 1953)

```
# I. Load historical immigration data
mig <- read.csv("Auxiliary/Brazil_Immigration_1884-1953.csv")

# II. Calculate total immigration by country of origin
mig <- mig %>%
  summarise(Germany = sum(Germans),
```

```r
            Spain = sum(Spaniards),
            Italy = sum(Italians),
            Japan = sum(Japanese),
            Portugal = sum(Portuguese),
            Ukraine = sum(Russians),
            Syria = sum(Syrians_Turks)/2,
            Lebanon = sum(Syrians_Turks)/2,
            Others = sum(Others),
            Total = sum(Total)) %>%
  pivot_longer(cols = everything(),
               names_to = "country",
               values_to = "values")

# III. Extract total for normalization
total <- mig %>%
  filter(country == "Total") %>%
  pull(values) %>%
  as.numeric()

# IV. Calculate cultural proximity index
mig <- mig %>%
  filter(country != "Total" & country != "Others") %>%
  mutate(index = case_when(country == "Germany" ~ 405861/total,
                           country == "Spain" ~ 606095/total,
                           country == "Italy" ~ 1515387/total,
                           country == "Japan" ~ 194221/total,
                           country == "Portugal" ~ 1401317/total,
                           country == "Ukraine" ~ 90079/total,
                           country == "Syria" ~ 48953/total,
                           country == "Lebanon" ~ 48953/total,
                           TRUE ~ 585354/total)) %>%
  rename(cont = country)

# V. Add cultural proximity to BRA dataset
BRA <- BRA %>%
  left_join(mig, by = "cont") %>%
  select(-values) %>%
  replace_na(list(index = 0)) %>%
  rename(cult_prox = index)

rm(mig)
```

3.4. Geographical Distance from Brazil (by closest border/shoreline distance)

```r
# I. Enable geodesic distances on the sphere
sf_use_s2(TRUE)

# II. Load world geometries
world <- ne_countries(scale = "medium", returnclass = "sf") %>%
  st_make_valid()
```

```
# III. Get Brazil geometry
bra <- world %>%
  filter(iso_a3 == "BRA") %>% st_union()

# IV. Function to calculate distance to Brazil
dist_to_brazil <- function(country_name_or_iso3) {
  x <- world %>%
    filter(admin == country_name_or_iso3 | name_long == country_name_or_iso3 | iso_a3 == country_name_o
    st_union()
  as.numeric(set_units(st_distance(x, bra), "km"))
}

# V. Get target countries
targets <- c(unique(BRA$cont), "North Macedonia", "Democratic Republic of the Congo")

# VI. Calculate distances
dist <- tibble(cont = targets,
               geo_distance = sapply(targets, dist_to_brazil)) %>%
  filter(!is.na(geo_distance)) %>%
  mutate(cont = case_when(cont == "North Macedonia" ~ "Macedonia",
                          cont == "Democratic Republic of the Congo" ~ "Dem. Rep. Congo",
                          TRUE ~ cont))

# VII. Add geographical distance to BRA dataset
BRA <- BRA %>%
  left_join(dist, by = "cont") %>%
  filter(!is.na(geo_distance))

rm(bra, dist, world)
```

3.5. Baseline mobility friction (based on entry visa requirement)

```
# I. Define years of interest
seq <- seq(1973, 2008)

# II. Convert to character for column selection
ycols <- as.character(seq)

# III. Load and process visa data
visa <- read_excel(path = "Auxiliary/DEMIG_VISA_Database_version_1.4.xlsx") %>%
  rename(cont = `Country of visa issuance`,
         bra_cont = `Nationality of traveller`,
         contcod = `UN 3-digit code...3`,
         policy = `Policy measure`) %>%
  select(c(cont, contcod, bra_cont, policy, all_of(as.character(seq)))) %>%
  filter(bra_cont == "Brazil" & policy == "Visa") %>%
  mutate(visa = rowMeans(across(all_of(ycols)), na.rm = TRUE)) %>%
  select(c(contcod, visa))
```

```
## New names:
## * ‘UN numeric code‘ -> ‘UN numeric code...2‘
## * ‘UN 3-digit code‘ -> ‘UN 3-digit code...3‘
## * ‘UN numeric code‘ -> ‘UN numeric code...6‘
## * ‘UN 3-digit code‘ -> ‘UN 3-digit code...7‘
```

```
# IV. Add visa requirement to BRA dataset
BRA <- BRA %>%
  left_join(visa, by = "contcod") %>%
  filter(!cont %in% c("South Sudan", "Vanuatu"))

rm(visa)
```

3.6. General policy restrictiveness at destination (0 = liberal, 1 = restrictive)

```
# I. Load IMPIC policy data
policy <- read_dta("Auxiliary/IMPICDatasetV2_1980-2018.dta")

# II. Get column names for labor and family policies
cols_a <- grep("^AvgS_a", names(policy), value = TRUE)

cols_b <- grep("^AvgS_b", names(policy), value = TRUE)

# III. Process policy data
policy <- policy %>%
  filter(year <= 2008 & year >= 2005) %>%
  mutate(cntry = countrycode(cntry, "iso2c", "iso3c", warn = FALSE)) %>%
  filter(!is.na(cntry)) %>%
  rename(contcod = cntry) %>%
  rowwise() %>%
  mutate(policy_labour = mean(c_across(all_of(cols_a)), na.rm = TRUE),
         policy_family = mean(c_across(all_of(cols_b)), na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(mig_policy = (policy_labour + policy_family) / 2) %>%
  select(c(contcod, year, mig_policy)) %>%
  group_by(contcod) %>%
  summarise(mig_policy = mean(mig_policy, na.rm = TRUE)) %>%
  filter(mig_policy != "NaN")

# IV. Add migration policy to BRA dataset
BRA <- BRA %>%
  left_join(policy, by = "contcod") %>%
  mutate(mig_policy = case_when(is.na(mig_policy) == TRUE ~ 0.5*visa,
                                TRUE ~ mig_policy))

rm(country_mapping, policy)
```

3.7. Network effects (bigger Brazilian diaspora in country j lowers costs such as info, housing, job search, etc)

```
# I. Retrieve data
raw <- read_excel(path = "Auxiliary/undesa_pd_2024_ims_stock_by_sex_destination_and_origin.xlsx",
```

```r
                   sheet = "Table 1")

# II. Keep rows where origin == Brazil
bra <- raw %>%
  filter(`Region, development group, country or area of origin` == "Brazil") %>%
  transmute(dest_m49 = `Location code of destination`,
            destination = `Region, development group, country or area of destination`,
            br_born_2000 = `International migrant stock (2000)`)

# III. Keep countries only (drop World/regions) and drop Brazil itself
# In UN M49, country codes are < 900; region aggregates are >= 900.
bra_countries <- bra %>%
  filter(dest_m49 < 900, dest_m49 != 76) %>%   # 76 = Brazil (M49)
  mutate(destination = str_trim(str_remove_all(destination, "\\*")))

# IV. Build the network variables
scale01 <- function(x) (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))

network <- bra_countries %>%
  mutate(br_log_2000 = log1p(br_born_2000), # Log transform to tame very large values
         network_z = scale01(br_log_2000), # Standardized log stock (network index)
         contcod = countrycode(destination, "country.name", "iso3c", warn = FALSE)) %>%
  filter(!is.na(contcod)) %>%
  select(-c(dest_m49, destination)) %>%
  relocate(contcod, .before = "br_born_2000")

# V. Add network effects to BRA dataset
BRA <- BRA %>%
  left_join(network, by = "contcod") %>%
  select(-c(br_born_2000, network_z)) %>%
  mutate(br_log_2000 = case_when(is.na(br_log_2000) ~ 0,
                                 TRUE ~ br_log_2000)) %>%
  rename(ln_diaspora = br_log_2000)

rm(bra, bra_countries, network, raw)
```

3.8. Political instability (as the number of coups/attempts between 1998 and 2008)

```r
# I. Load coup data
coup <- read.csv(file = "Auxiliary/Coup_data_2.2.0.csv") %>%
  filter(year >= 1998 & year <= 2008) %>%
  mutate(contcod = countrycode(country, "country.name", "iso3c", warn = FALSE)) %>%
  select(c(contcod, year, event_type)) %>%
  relocate(contcod, .before = year) %>%
  group_by(contcod) %>%
  summarise(n_coups = n()) # Number of coups, attempted coups, and conspiracies

# II. Add political instability to BRA dataset
BRA <- BRA %>%
  left_join(coup, by = "contcod") %>%
  mutate(n_coups = case_when(is.na(n_coups) ~ 0,
                             TRUE ~ n_coups))

rm(coup)
```

3.9. Gravity model (Poisson Pseudo Maximum Likelihood - PPML)

```
# I. Load income and inequality data
WYD <- read_excel(path = "Datasets/WYD_reg.xlsx") %>%
  group_by(contcod) %>%
  summarise(lninc = weighted.mean(lninc, w = pop, na.rm = TRUE),
            gini = dplyr::first(gini[!is.na(gini)]),
            pop = mean(pop))

# II. Prepare final dataset for gravity model
gravity_df <- BRA %>%
  left_join(WYD, by = "contcod") %>%
  filter(!is.na(lninc)) %>%
  mutate(ln_dist = log(geo_distance),
         ln_inc = as.numeric(lninc)) %>%
  rename(cult_dist = cult_prox,
         language_dist = language_distance,
         geo_dist = geo_distance) %>%
  select(-lninc)

save(gravity_df,
     file = "Datasets/gravity_df.RData")

load("Datasets/gravity_df.RData")

# III. Estimate gravity model with Poisson Pseudo Maximum Likelihood
gravity_model <- fepois(data = gravity_df,
                        expats ~ ln_dist + ln_inc + language_dist + cult_dist + gini + visa + mig_polic
                        offset = log(gravity_df$pop),
                        vcov = "hetero",
                        fixef = "reg")
```

```
## NOTES: 8 observations removed because of infinite values (RHS: 8).
##        1 fixed-effect (1 observation) removed because of only 0 outcomes.
```

```
# IV. Display model results
summary(gravity_model)
```

```
## Poisson estimation, Dep. Var.: expats
## Observations: 101
## Offset: log(gravity_df$pop)
## Fixed-effects: reg: 6
## Standard-errors: Heteroskedasticity-robust
##                Estimate Std. Error   z value   Pr(>|z|)
## ln_dist        0.064695   0.277448  0.233178 8.1562e-01
## ln_inc         2.867749   0.472295   6.071942 1.2637e-09 ***
## language_dist -4.734224   0.387570 -12.215132  < 2.2e-16 ***
## cult_dist     -3.217388   1.287181  -2.499562 1.2435e-02 *
## gini           9.424121   3.920928   2.403543 1.6237e-02 *
## visa          -0.166790   0.576501  -0.289314 7.7234e-01
## mig_policy     0.480162   1.847936   0.259837 7.9499e-01
## ln_diaspora    0.141456   0.041810   3.383278 7.1626e-04 ***
## n_coups        0.400908   0.326322   1.228565 2.1923e-01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Log-Likelihood: -200,702.0   Adj. Pseudo R2: 0.969713
##            BIC:  401,473.2     Squared Cor.: 0.993294
```