FA2_Patayon

PATAYON, SPIKE LEE-ROY V

2026-02-02

1. Introduction

In this formative assessment, you will review Unit 1, focusing on data wrangling, manipulation, and visualization.

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

You will perform a data analysis to study trends in tuberculosis (TB) cases worldwide over time. The two relevant datasets are:

- **`who.tsv`** : Information about TB cases in various countries from 1980 to 2013 (Source: 2014 WHO Global Tuberculosis Report).
- **`population.csv`** : Population data of each country across time (Source: The World Bank).

2. Import

2.1. Preview the contents of **`who.tsv`** and **`population.csv`** by inspecting the files.

2.2. Import the data into tibbles named `who` and `population` .

```
df_who = who

df_population = read_csv("C:\\Users\\spike\\Downloads\\Population.csv")
```

```
## Rows: 266 Columns: 68
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): Country Name, Country Code, Indicator Name, Indicator Code
## dbl (64): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
##
## 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.
```

```
df_who
```

```
## # A tibble: 7,240 × 60
##    country  iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##    <chr>    <chr> <chr> <dbl>       <dbl>        <dbl>        <dbl>        <dbl>
##  1 Afghani… AF    AFG    1980          NA           NA           NA           NA
##  2 Afghani… AF    AFG    1981          NA           NA           NA           NA
##  3 Afghani… AF    AFG    1982          NA           NA           NA           NA
##  4 Afghani… AF    AFG    1983          NA           NA           NA           NA
##  5 Afghani… AF    AFG    1984          NA           NA           NA           NA
##  6 Afghani… AF    AFG    1985          NA           NA           NA           NA
##  7 Afghani… AF    AFG    1986          NA           NA           NA           NA
##  8 Afghani… AF    AFG    1987          NA           NA           NA           NA
##  9 Afghani… AF    AFG    1988          NA           NA           NA           NA
## 10 Afghani… AF    AFG    1989          NA           NA           NA           NA
## # ℹ 7,230 more rows
## # ℹ 52 more variables: new_sp_m4554 <dbl>, new_sp_m5564 <dbl>,
## #   new_sp_m65 <dbl>, new_sp_f014 <dbl>, new_sp_f1524 <dbl>,
## #   new_sp_f2534 <dbl>, new_sp_f3544 <dbl>, new_sp_f4554 <dbl>,
## #   new_sp_f5564 <dbl>, new_sp_f65 <dbl>, new_sn_m014 <dbl>,
## #   new_sn_m1524 <dbl>, new_sn_m2534 <dbl>, new_sn_m3544 <dbl>,
## #   new_sn_m4554 <dbl>, new_sn_m5564 <dbl>, new_sn_m65 <dbl>, …
```

```
df_population
```

```
## # A tibble: 266 × 68
##    `Country Name` `Country Code` `Indicator Name` `Indicator Code` `1960` `1961`
##    <chr>          <chr>          <chr>            <chr>             <dbl>  <dbl>
##  1 Aruba          ABW            Population, tot… SP.POP.TOTL      5.49e4 5.56e4
##  2 Africa Easter… AFE            Population, tot… SP.POP.TOTL      1.30e8 1.34e8
##  3 Afghanistan    AFG            Population, tot… SP.POP.TOTL      9.04e6 9.21e6
##  4 Africa Wester… AFW            Population, tot… SP.POP.TOTL      9.76e7 9.97e7
##  5 Angola         AGO            Population, tot… SP.POP.TOTL      5.23e6 5.30e6
##  6 Albania        ALB            Population, tot… SP.POP.TOTL      1.61e6 1.66e6
##  7 Andorra        AND            Population, tot… SP.POP.TOTL      9.51e3 1.03e4
##  8 Arab World     ARB            Population, tot… SP.POP.TOTL      9.15e7 9.39e7
##  9 United Arab E… ARE            Population, tot… SP.POP.TOTL      1.31e5 1.38e5
## 10 Argentina      ARG            Population, tot… SP.POP.TOTL      2.04e7 2.07e7
## # ℹ 256 more rows
## # ℹ 62 more variables: `1962` <dbl>, `1963` <dbl>, `1964` <dbl>, `1965` <dbl>,
## #   `1966` <dbl>, `1967` <dbl>, `1968` <dbl>, `1969` <dbl>, `1970` <dbl>,
## #   `1971` <dbl>, `1972` <dbl>, `1973` <dbl>, `1974` <dbl>, `1975` <dbl>,
## #   `1976` <dbl>, `1977` <dbl>, `1978` <dbl>, `1979` <dbl>, `1980` <dbl>,
## #   `1981` <dbl>, `1982` <dbl>, `1983` <dbl>, `1984` <dbl>, `1985` <dbl>,
## #   `1986` <dbl>, `1987` <dbl>, `1988` <dbl>, `1989` <dbl>, `1990` <dbl>, …
```

2.3. Determine the number of rows and columns in each tibble.

```
dim(df_who)
```

```
## [1] 7240   60
```

```
dim(df_population)
```

```
## [1] 266  68
```

2.4. Check the summary of variable types for `population.csv`. Fix any anomalies and store the corrected data in `population2`.

```
glimpse(df_population)
```

```
## Rows: 266
## Columns: 68
## $ `Country Name`   <chr> "Aruba", "Africa Eastern and Southern", "Afghanistan"…
## $ `Country Code`   <chr> "ABW", "AFE", "AFG", "AFW", "AGO", "ALB", "AND", "ARB…
## $ `Indicator Name` <chr> "Population, total", "Population, total", "Population…
## $ `Indicator Code` <chr> "SP.POP.TOTL", "SP.POP.TOTL", "SP.POP.TOTL", "SP.POP.…
## $ `1960`           <dbl> 54922, 130072080, 9035043, 97630925, 5231654, 1608800…
## $ `1961`           <dbl> 55578, 133534923, 9214083, 99706674, 5301583, 1659800…
## $ `1962`           <dbl> 56320, 137171659, 9404406, 101854756, 5354310, 171131…
## $ `1963`           <dbl> 57002, 140945536, 9604487, 104089175, 5408320, 176262…
## $ `1964`           <dbl> 57619, 144904094, 9814318, 106388440, 5464187, 181413…
## $ `1965`           <dbl> 58190, 149033472, 10036008, 108772632, 5521981, 18647…
## $ `1966`           <dbl> 58694, 153281203, 10266395, 111246953, 5581386, 19145…
## $ `1967`           <dbl> 58990, 157704381, 10505959, 113795019, 5641807, 19655…
## $ `1968`           <dbl> 59069, 162329396, 10756922, 116444636, 5702699, 20222…
## $ `1969`           <dbl> 59052, 167088245, 11017409, 119203521, 5763685, 20816…
## $ `1970`           <dbl> 58950, 171984985, 11290128, 122086536, 5852788, 21354…
## $ `1971`           <dbl> 58781, 177022314, 11567667, 125072948, 5991102, 21878…
## $ `1972`           <dbl> 58047, 182126556, 11853696, 128176494, 6174262, 22431…
## $ `1973`           <dbl> 58299, 187524135, 12157999, 131449942, 6388528, 22967…
## $ `1974`           <dbl> 58349, 193186642, 12469127, 134911581, 6613367, 23501…
## $ `1975`           <dbl> 58295, 198914573, 12773954, 138569918, 6842947, 24048…
## $ `1976`           <dbl> 58368, 204802976, 13059851, 142337272, 7074664, 24585…
## $ `1977`           <dbl> 58580, 210680842, 13340756, 146258576, 7317829, 25135…
## $ `1978`           <dbl> 58776, 217074286, 13611441, 150402616, 7576734, 25662…
## $ `1979`           <dbl> 59191, 223974122, 13655567, 154721711, 7847207, 26178…
## $ `1980`           <dbl> 59909, 230792729, 13169311, 159166518, 8133872, 26719…
## $ `1981`           <dbl> 60563, 238043099, 11937581, 163762473, 8435607, 27260…
## $ `1982`           <dbl> 61276, 245822010, 10991378, 168585118, 8751648, 27842…
## $ `1983`           <dbl> 62228, 253644643, 10917982, 173255157, 9082983, 28439…
## $ `1984`           <dbl> 62901, 261458202, 11190221, 177880746, 9425917, 29044…
## $ `1985`           <dbl> 61728, 269450407, 11426852, 182811038, 9779120, 29647…
## $ `1986`           <dbl> 59931, 277621771, 11420074, 187889141, 10139450, 3022…
## $ `1987`           <dbl> 59159, 286067346, 11387818, 193104347, 10497858, 3083…
## $ `1988`           <dbl> 59331, 294498625, 11523298, 198485027, 10861291, 3142…
## $ `1989`           <dbl> 60443, 302939121, 11874088, 204062274, 11238562, 3227…
## $ `1990`           <dbl> 62753, 311748681, 12045660, 209566031, 11626360, 3286…
## $ `1991`           <dbl> 65896, 320442961, 12238879, 215178709, 12023529, 3266…
## $ `1992`           <dbl> 69005, 329082707, 13278974, 221191375, 12423712, 3247…
## $ `1993`           <dbl> 73685, 338324002, 14943172, 227246778, 12827135, 3227…
## $ `1994`           <dbl> 77595, 347441809, 16250794, 233360104, 13249764, 3207…
## $ `1995`           <dbl> 79805, 356580375, 17065836, 239801875, 13699778, 3187…
## $ `1996`           <dbl> 83021, 366138524, 17763266, 246415446, 14170973, 3168…
## $ `1997`           <dbl> 86301, 375646235, 18452091, 253207584, 14660413, 3148…
## $ `1998`           <dbl> 88451, 385505757, 19159996, 260297834, 15159370, 3128…
## $ `1999`           <dbl> 89659, 395750933, 19887785, 267506298, 15667235, 3108…
## $ `2000`           <dbl> 90588, 406156661, 20130327, 274968446, 16194869, 3089…
## $ `2001`           <dbl> 91439, 416807868, 20284307, 282780717, 16747208, 3060…
## $ `2002`           <dbl> 92074, 427820358, 21378117, 290841795, 17327699, 3051…
## $ `2003`           <dbl> 93128, 439173286, 22733049, 299142845, 17943712, 3039…
## $ `2004`           <dbl> 95138, 450928044, 23560654, 307725100, 18600423, 3026…
## $ `2005`           <dbl> 97635, 463076637, 24404567, 316588476, 19291161, 3011…
## $ `2006`           <dbl> 99405, 475606210, 25424094, 325663158, 20015279, 2992…
## $ `2007`           <dbl> 100150, 488580707, 25909852, 334984176, 20778561, 297…
## $ `2008`           <dbl> 100917, 502070763, 26482622, 344586109, 21578655, 294…
## $ `2009`           <dbl> 101604, 516003448, 27466101, 354343844, 22414773, 292…
```

```
## $ `2010`           <dbl> 101838, 530308387, 28284089, 364358270, 23294825, 291…
## $ `2011`           <dbl> 102591, 544737983, 29347708, 374790143, 24218352, 290…
## $ `2012`           <dbl> 104110, 559609961, 30560034, 385360349, 25177394, 290…
## $ `2013`           <dbl> 105675, 575202699, 31622704, 396030207, 26165620, 289…
## $ `2014`           <dbl> 106807, 590968990, 32792523, 406992047, 27160769, 288…
## $ `2015`           <dbl> 107906, 607123269, 33831764, 418127845, 28157798, 288…
## $ `2016`           <dbl> 108727, 623369401, 34700612, 429454743, 29183070, 287…
## $ `2017`           <dbl> 108735, 640058741, 35688935, 440882906, 30234839, 287…
## $ `2018`           <dbl> 108908, 657801085, 36743039, 452195915, 31297155, 286…
## $ `2019`           <dbl> 109203, 675950189, 37856121, 463365429, 32375632, 285…
## $ `2020`           <dbl> 108587, 694446100, 39068979, 474569351, 33451132, 283…
## $ `2021`           <dbl> 107700, 713090928, 40000412, 485920997, 34532429, 281…
## $ `2022`           <dbl> 107310, 731821393, 40578842, 497387180, 35635029, 277…
## $ `2023`           <dbl> 107359, 750503764, 41454761, 509398589, 36749906, 274…
```

```
summary(df_population)
```

```
##  Country Name       Country Code       Indicator Name     Indicator Code
##  Length:266         Length:266         Length:266         Length:266
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       1960               1961               1962
##  Min.   :2.715e+03   Min.   :2.970e+03   Min.   :3.264e+03
##  1st Qu.:5.152e+05   1st Qu.:5.255e+05   1st Qu.:5.363e+05
##  Median :3.660e+06   Median :3.747e+06   Median :3.832e+06
##  Mean   :1.154e+08   Mean   :1.171e+08   Mean   :1.192e+08
##  3rd Qu.:2.686e+07   3rd Qu.:2.761e+07   3rd Qu.:2.837e+07
##  Max.   :3.022e+09   Max.   :3.063e+09   Max.   :3.117e+09
##  NA's   :2           NA's   :2           NA's   :2
##       1963               1964               1965
##  Min.   :3.584e+03   Min.   :3.922e+03   Min.   :4.282e+03
##  1st Qu.:5.476e+05   1st Qu.:5.594e+05   1st Qu.:5.676e+05
##  Median :3.920e+06   Median :4.010e+06   Median :4.103e+06
##  Mean   :1.219e+08   Mean   :1.246e+08   Mean   :1.273e+08
##  3rd Qu.:2.915e+07   3rd Qu.:2.995e+07   3rd Qu.:3.076e+07
##  Max.   :3.184e+09   Max.   :3.251e+09   Max.   :3.319e+09
##  NA's   :2           NA's   :2           NA's   :2
##       1966               1967               1968
##  Min.   :4.664e+03   Min.   :5.071e+03   Min.   :5.500e+03
##  1st Qu.:5.712e+05   1st Qu.:5.780e+05   1st Qu.:5.825e+05
##  Median :4.199e+06   Median :4.298e+06   Median :4.396e+06
##  Mean   :1.302e+08   Mean   :1.330e+08   Mean   :1.359e+08
##  3rd Qu.:3.148e+07   3rd Qu.:3.204e+07   3rd Qu.:3.247e+07
##  Max.   :3.389e+09   Max.   :3.459e+09   Max.   :3.531e+09
##  NA's   :2           NA's   :2           NA's   :2
##       1969               1970               1971
##  Min.   :5.631e+03   Min.   :5.663e+03   Min.   :5.685e+03
##  1st Qu.:5.861e+05   1st Qu.:5.945e+05   1st Qu.:6.064e+05
##  Median :4.503e+06   Median :4.564e+06   Median :4.607e+06
##  Mean   :1.390e+08   Mean   :1.421e+08   Mean   :1.452e+08
##  3rd Qu.:3.277e+07   3rd Qu.:3.295e+07   3rd Qu.:3.322e+07
##  Max.   :3.605e+09   Max.   :3.681e+09   Max.   :3.759e+09
##  NA's   :2           NA's   :2           NA's   :2
##       1972               1973               1974
##  Min.   :5.692e+03   Min.   :5.681e+03   Min.   :5.822e+03
##  1st Qu.:6.220e+05   1st Qu.:6.478e+05   1st Qu.:6.594e+05
##  Median :4.697e+06   Median :4.834e+06   Median :4.958e+06
##  Mean   :1.483e+08   Mean   :1.515e+08   Mean   :1.546e+08
##  3rd Qu.:3.378e+07   3rd Qu.:3.432e+07   3rd Qu.:3.486e+07
##  Max.   :3.834e+09   Max.   :3.911e+09   Max.   :3.987e+09
##  NA's   :2           NA's   :2           NA's   :2
##       1975               1976               1977
##  Min.   :6.117e+03   Min.   :6.404e+03   Min.   :6.686e+03
##  1st Qu.:6.615e+05   1st Qu.:6.900e+05   1st Qu.:7.339e+05
##  Median :5.053e+06   Median :5.162e+06   Median :5.301e+06
##  Mean   :1.577e+08   Mean   :1.607e+08   Mean   :1.637e+08
##  3rd Qu.:3.540e+07   3rd Qu.:3.592e+07   3rd Qu.:3.644e+07
##  Max.   :4.062e+09   Max.   :4.136e+09   Max.   :4.209e+09
##  NA's   :2           NA's   :2           NA's   :2
```

```
##       1978                1979                1980          
##  Min.   :6.957e+03   Min.   :7.184e+03   Min.   :7.366e+03  
##  1st Qu.:7.849e+05   1st Qu.:8.074e+05   1st Qu.:8.156e+05  
##  Median :5.392e+06   Median :5.549e+06   Median :5.744e+06  
##  Mean   :1.668e+08   Mean   :1.700e+08   Mean   :1.732e+08  
##  3rd Qu.:3.701e+07   3rd Qu.:3.776e+07   3rd Qu.:3.855e+07  
##  Max.   :4.283e+09   Max.   :4.360e+09   Max.   :4.438e+09  
##  NA's   :2           NA's   :2           NA's   :2          
##       1981                1982                1983          
##  Min.   :7.526e+03   Min.   :7.664e+03   Min.   :7.789e+03  
##  1st Qu.:8.259e+05   1st Qu.:8.392e+05   1st Qu.:8.549e+05  
##  Median :5.878e+06   Median :6.005e+06   Median :6.148e+06  
##  Mean   :1.765e+08   Mean   :1.800e+08   Mean   :1.835e+08  
##  3rd Qu.:3.954e+07   3rd Qu.:4.056e+07   3rd Qu.:4.144e+07  
##  Max.   :4.517e+09   Max.   :4.600e+09   Max.   :4.683e+09  
##  NA's   :2           NA's   :2           NA's   :2          
##       1984                1985                1986          
##  Min.   :7.902e+03   Min.   :8.046e+03   Min.   :8.217e+03  
##  1st Qu.:8.710e+05   1st Qu.:8.880e+05   1st Qu.:9.076e+05  
##  Median :6.285e+06   Median :6.423e+06   Median :6.542e+06  
##  Mean   :1.869e+08   Mean   :1.905e+08   Mean   :1.942e+08  
##  3rd Qu.:4.225e+07   3rd Qu.:4.300e+07   3rd Qu.:4.370e+07  
##  Max.   :4.766e+09   Max.   :4.851e+09   Max.   :4.938e+09  
##  NA's   :2           NA's   :2           NA's   :2          
##       1987                1988                1989          
##  Min.   :8.371e+03   Min.   :8.518e+03   Min.   :8.662e+03  
##  1st Qu.:9.300e+05   1st Qu.:9.631e+05   1st Qu.:9.980e+05  
##  Median :6.702e+06   Median :6.846e+06   Median :7.038e+06  
##  Mean   :1.979e+08   Mean   :2.017e+08   Mean   :2.055e+08  
##  3rd Qu.:4.472e+07   3rd Qu.:4.604e+07   3rd Qu.:4.739e+07  
##  Max.   :5.027e+09   Max.   :5.117e+09   Max.   :5.208e+09  
##  NA's   :2           NA's   :2           NA's   :2          
##       1990                1991                1992          
##  Min.   :8.798e+03   Min.   :8.928e+03   Min.   :9.038e+03  
##  1st Qu.:1.055e+06   1st Qu.:1.070e+06   1st Qu.:1.084e+06  
##  Median :7.130e+06   Median :7.271e+06   Median :7.382e+06  
##  Mean   :2.085e+08   Mean   :2.123e+08   Mean   :2.160e+08  
##  3rd Qu.:4.761e+07   3rd Qu.:4.972e+07   3rd Qu.:5.185e+07  
##  Max.   :5.299e+09   Max.   :5.388e+09   Max.   :5.477e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       1993                1994                1995          
##  Min.   :9.126e+03   Min.   :9.207e+03   Min.   :9.280e+03  
##  1st Qu.:1.097e+06   1st Qu.:1.113e+06   1st Qu.:1.123e+06  
##  Median :7.495e+06   Median :7.420e+06   Median :7.563e+06  
##  Mean   :2.197e+08   Mean   :2.233e+08   Mean   :2.269e+08  
##  3rd Qu.:5.235e+07   3rd Qu.:5.208e+07   3rd Qu.:5.167e+07  
##  Max.   :5.565e+09   Max.   :5.651e+09   Max.   :5.737e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       1996                1997                1998          
##  Min.   :9.342e+03   Min.   :9.400e+03   Min.   :9.451e+03  
##  1st Qu.:1.152e+06   1st Qu.:1.181e+06   1st Qu.:1.212e+06  
##  Median :7.708e+06   Median :7.838e+06   Median :8.002e+06  
##  Mean   :2.306e+08   Mean   :2.342e+08   Mean   :2.378e+08  
##  3rd Qu.:5.123e+07   3rd Qu.:5.079e+07   3rd Qu.:5.038e+07  
##  Max.   :5.823e+09   Max.   :5.908e+09   Max.   :5.994e+09  
##  NA's   :1           NA's   :1           NA's   :1
```

```
##       1999                2000                2001          
##  Min.   :9.496e+03   Min.   :9.544e+03   Min.   :9.586e+03  
##  1st Qu.:1.243e+06   1st Qu.:1.276e+06   1st Qu.:1.310e+06  
##  Median :8.151e+06   Median :8.214e+06   Median :8.287e+06  
##  Mean   :2.414e+08   Mean   :2.450e+08   Mean   :2.485e+08  
##  3rd Qu.:4.998e+07   3rd Qu.:5.051e+07   3rd Qu.:5.213e+07  
##  Max.   :6.078e+09   Max.   :6.162e+09   Max.   :6.245e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       2002                2003                2004          
##  Min.   :9.623e+03   Min.   :9.695e+03   Min.   :9.816e+03  
##  1st Qu.:1.332e+06   1st Qu.:1.339e+06   1st Qu.:1.350e+06  
##  Median :8.400e+06   Median :8.587e+06   Median :8.817e+06  
##  Mean   :2.521e+08   Mean   :2.556e+08   Mean   :2.592e+08  
##  3rd Qu.:5.375e+07   3rd Qu.:5.534e+07   3rd Qu.:5.700e+07  
##  Max.   :6.328e+09   Max.   :6.410e+09   Max.   :6.493e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       2005                2006                2007          
##  Min.   :9.940e+03   Min.   :1.003e+04   Min.   :1.002e+04  
##  1st Qu.:1.355e+06   1st Qu.:1.347e+06   1st Qu.:1.341e+06  
##  Median :9.030e+06   Median :9.081e+06   Median :9.148e+06  
##  Mean   :2.628e+08   Mean   :2.664e+08   Mean   :2.700e+08  
##  3rd Qu.:5.797e+07   3rd Qu.:5.814e+07   3rd Qu.:5.844e+07  
##  Max.   :6.576e+09   Max.   :6.660e+09   Max.   :6.744e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       2008                2009                2010          
##  Min.   :1.001e+04   Min.   :1.002e+04   Min.   :1.004e+04  
##  1st Qu.:1.427e+06   1st Qu.:1.526e+06   1st Qu.:1.566e+06  
##  Median :9.228e+06   Median :9.505e+06   Median :9.746e+06  
##  Mean   :2.737e+08   Mean   :2.774e+08   Mean   :2.812e+08  
##  3rd Qu.:5.883e+07   3rd Qu.:5.910e+07   3rd Qu.:5.928e+07  
##  Max.   :6.830e+09   Max.   :6.916e+09   Max.   :7.001e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       2011                2012                2013          
##  Min.   :1.010e+04   Min.   :1.027e+04   Min.   :1.052e+04  
##  1st Qu.:1.608e+06   1st Qu.:1.651e+06   1st Qu.:1.696e+06  
##  Median :9.915e+06   Median :1.007e+07   Median :1.019e+07  
##  Mean   :2.849e+08   Mean   :2.888e+08   Mean   :2.927e+08  
##  3rd Qu.:5.938e+07   3rd Qu.:5.954e+07   3rd Qu.:6.023e+07  
##  Max.   :7.086e+09   Max.   :7.176e+09   Max.   :7.265e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       2014                2015                2016          
##  Min.   :1.074e+04   Min.   :1.095e+04   Min.   :1.093e+04  
##  1st Qu.:1.741e+06   1st Qu.:1.786e+06   1st Qu.:1.778e+06  
##  Median :1.032e+07   Median :1.036e+07   Median :1.033e+07  
##  Mean   :2.966e+08   Mean   :3.005e+08   Mean   :3.043e+08  
##  3rd Qu.:6.079e+07   3rd Qu.:6.073e+07   3rd Qu.:6.063e+07  
##  Max.   :7.354e+09   Max.   :7.441e+09   Max.   :7.529e+09  
##  NA's   :1           NA's   :1           NA's   :1          
##       2017                2018                2019          
##  Min.   :1.087e+04   Min.   :1.075e+04   Min.   :1.058e+04  
##  1st Qu.:1.791e+06   1st Qu.:1.797e+06   1st Qu.:1.789e+06  
##  Median :1.026e+07   Median :1.028e+07   Median :1.042e+07  
##  Mean   :3.082e+08   Mean   :3.119e+08   Mean   :3.156e+08  
##  3rd Qu.:6.054e+07   3rd Qu.:6.042e+07   3rd Qu.:5.973e+07  
##  Max.   :7.614e+09   Max.   :7.696e+09   Max.   :7.777e+09  
##  NA's   :1           NA's   :1           NA's   :1
```

```
##       2020                2021                2022
##  Min.   :1.040e+04   Min.   :1.019e+04   Min.   :9.992e+03
##  1st Qu.:1.790e+06   1st Qu.:1.786e+06   1st Qu.:1.804e+06
##  Median :1.070e+07   Median :1.051e+07   Median :1.049e+07
##  Mean   :3.192e+08   Mean   :3.224e+08   Mean   :3.255e+08
##  3rd Qu.:6.097e+07   3rd Qu.:6.283e+07   3rd Qu.:6.471e+07
##  Max.   :7.856e+09   Max.   :7.921e+09   Max.   :7.990e+09
##  NA's   :1           NA's   :1           NA's   :1
##       2023
##  Min.   :9.816e+03
##  1st Qu.:1.828e+06
##  Median :1.064e+07
##  Mean   :3.288e+08
##  3rd Qu.:6.662e+07
##  Max.   :8.062e+09
##  NA's   :1
```

- What the summary tells us
    - The dataset has 266 rows and 68 columns
- Key identifier variables:
    - Country Name – character
    - Country Code – character
    - Indicator Name – character
    - Indicator Code – character
- Year columns (1960 to 2023) are correctly read as numeric ()
- Some year columns contain NA values (usually for aggregates like regions)

Rename columns for consistency

```
population2 <- df_population %>%
  rename(
    country = `Country Name`,
    country_code = `Country Code`
  ) %>%
  select(-`Indicator Name`, -`Indicator Code`)
```

Remove unnecessary indicator columns

```
population2
```

```
## # A tibble: 266 × 66
##    country  country_code `1960` `1961` `1962` `1963` `1964` `1965` `1966` `1967`
##    <chr>    <chr>         <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 Aruba    ABW          5.49e4 5.56e4 5.63e4 5.70e4 5.76e4 5.82e4 5.87e4 5.90e4
##  2 Africa … AFE          1.30e8 1.34e8 1.37e8 1.41e8 1.45e8 1.49e8 1.53e8 1.58e8
##  3 Afghani… AFG          9.04e6 9.21e6 9.40e6 9.60e6 9.81e6 1.00e7 1.03e7 1.05e7
##  4 Africa … AFW          9.76e7 9.97e7 1.02e8 1.04e8 1.06e8 1.09e8 1.11e8 1.14e8
##  5 Angola   AGO          5.23e6 5.30e6 5.35e6 5.41e6 5.46e6 5.52e6 5.58e6 5.64e6
##  6 Albania  ALB          1.61e6 1.66e6 1.71e6 1.76e6 1.81e6 1.86e6 1.91e6 1.97e6
##  7 Andorra  AND          9.51e3 1.03e4 1.11e4 1.19e4 1.28e4 1.36e4 1.46e4 1.58e4
##  8 Arab Wo… ARB          9.15e7 9.39e7 9.64e7 9.90e7 1.02e8 1.04e8 1.07e8 1.10e8
##  9 United … ARE          1.31e5 1.38e5 1.45e5 1.52e5 1.60e5 1.67e5 1.74e5 1.80e5
## 10 Argenti… ARG          2.04e7 2.07e7 2.11e7 2.14e7 2.18e7 2.21e7 2.25e7 2.28e7
## # ℹ 256 more rows
## # ℹ 56 more variables: `1968` <dbl>, `1969` <dbl>, `1970` <dbl>, `1971` <dbl>,
## #   `1972` <dbl>, `1973` <dbl>, `1974` <dbl>, `1975` <dbl>, `1976` <dbl>,
## #   `1977` <dbl>, `1978` <dbl>, `1979` <dbl>, `1980` <dbl>, `1981` <dbl>,
## #   `1982` <dbl>, `1983` <dbl>, `1984` <dbl>, `1985` <dbl>, `1986` <dbl>,
## #   `1987` <dbl>, `1988` <dbl>, `1989` <dbl>, `1990` <dbl>, `1991` <dbl>,
## #   `1992` <dbl>, `1993` <dbl>, `1994` <dbl>, `1995` <dbl>, `1996` <dbl>, …
```

The summary shows that the population dataset contains 266 rows and 68 columns. The country identifiers are stored as character variables, while yearly population values from 1960 to 2023 are stored as numeric variables. However, the dataset is in wide format, with each year represented as a separate column, and it includes regional aggregates rather than only individual countries. Additionally, some population values are missing. These anomalies were addressed by renaming variables for consistency and removing unnecessary indicator columns, storing the corrected dataset as `population2`.

3. Tidy Data

3.1 `who` Dataset

Description of Columns:

- **`country`** : Country name.
- **`iso2`** : Two-digit country code.
- **`iso3`** : Three-digit country code.
- **`year`** : Year.
- Variables like **`new_ep_f014`** :
  - `ep` : TB type (e.g., rel = relapse, ep = extrapulmonary).
  - `f` : Sex (e.g., f = female).
  - `014` : Age group (e.g., 0-14 years).

Steps:

3.1.1 Identify the variables in the dataset.

```
colnames(df_who)
```

```
##  [1] "country"      "iso2"         "iso3"         "year"         "new_sp_m014"
##  [6] "new_sp_m1524" "new_sp_m2534" "new_sp_m3544" "new_sp_m4554" "new_sp_m5564"
## [11] "new_sp_m65"   "new_sp_f014"  "new_sp_f1524" "new_sp_f2534" "new_sp_f3544"
## [16] "new_sp_f4554" "new_sp_f5564" "new_sp_f65"   "new_sn_m014"  "new_sn_m1524"
## [21] "new_sn_m2534" "new_sn_m3544" "new_sn_m4554" "new_sn_m5564" "new_sn_m65"
## [26] "new_sn_f014"  "new_sn_f1524" "new_sn_f2534" "new_sn_f3544" "new_sn_f4554"
## [31] "new_sn_f5564" "new_sn_f65"   "new_ep_m014"  "new_ep_m1524" "new_ep_m2534"
## [36] "new_ep_m3544" "new_ep_m4554" "new_ep_m5564" "new_ep_m65"   "new_ep_f014"
## [41] "new_ep_f1524" "new_ep_f2534" "new_ep_f3544" "new_ep_f4554" "new_ep_f5564"
## [46] "new_ep_f65"   "newrel_m014"  "newrel_m1524" "newrel_m2534" "newrel_m3544"
## [51] "newrel_m4554" "newrel_m5564" "newrel_m65"   "newrel_f014"  "newrel_f1524"
## [56] "newrel_f2534" "newrel_f3544" "newrel_f4554" "newrel_f5564" "newrel_f65"
```

3.1.2 Perform a pivot operation to make the data tidy, storing the result in `who2`.

```
who2 <- who %>%
  pivot_longer(
    cols = starts_with("new"),
    names_to = "key",
    values_to = "cases"
  )
```

Remove NA values

```
who2 <- who2 %>%
  filter(!is.na(cases))
```

3.1.3 Separate values like `new_ep_f014` into components (e.g., `new`, `ep`, `f014`). Remove the column containing `new`, and store the result in `who3`.

```
who3 <- who2 %>%
  separate(key, into = c("new", "type", "sex_age"), sep = "_") %>%
  select(-new)
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2580 rows [243, 244, 679,
## 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 903, 904, 905,
## 906, ...].
```

3.1.4 Further separate values like `f014` into `f` and `014`, storing the result in `who_tidy`.

```
who_tidy <- who3 %>%
  separate(sex_age, into = c("sex", "age"), sep = 1)
```

3.2 Population Dataset

3.2.1 Identify the variables in this dataset.

```
colnames(population2)
```

```
##  [1] "country"      "country_code" "1960"         "1961"         "1962"
##  [6] "1963"         "1964"         "1965"         "1966"         "1967"
## [11] "1968"         "1969"         "1970"         "1971"         "1972"
## [16] "1973"         "1974"         "1975"         "1976"         "1977"
## [21] "1978"         "1979"         "1980"         "1981"         "1982"
## [26] "1983"         "1984"         "1985"         "1986"         "1987"
## [31] "1988"         "1989"         "1990"         "1991"         "1992"
## [36] "1993"         "1994"         "1995"         "1996"         "1997"
## [41] "1998"         "1999"         "2000"         "2001"         "2002"
## [46] "2003"         "2004"         "2005"         "2006"         "2007"
## [51] "2008"         "2009"         "2010"         "2011"         "2012"
## [56] "2013"         "2014"         "2015"         "2016"         "2017"
## [61] "2018"         "2019"         "2020"         "2021"         "2022"
## [66] "2023"
```

3.2.2 Perform a pivot operation to tidy the data, storing the result in `population3`.

```
population3 <- population2 %>%
  pivot_longer(
    cols = matches("^[0-9]{4}$"),
    names_to = "year",
    values_to = "population"
  )
```

3.2.3 Cast the population variable to an appropriate data type, storing the result in `population_tidy`.

```
population_tidy <- population3 %>%
  mutate(
    year = as.integer(year),
    population = as.numeric(population)
  )
```

3.3 Join Datasets

3.3.1 Identify the variable(s) required to join `who_tidy` and `population_tidy`.

```
colnames(who_tidy)
```

```
## [1] "country" "iso2"    "iso3"    "year"    "type"    "sex"     "age"
## [8] "cases"
```

```
colnames(population_tidy)
```

```
## [1] "country"      "country_code" "year"         "population"
```

3.3.2 Rename columns as needed to align variable names between datasets.

```
population_tidy <- population_tidy %>%
  rename(country = country)
```

3.3.3 Join the datasets into a tibble called `tuberculosis`.

```
tuberculosis <- who_tidy %>%
  left_join(population_tidy, by = c("country", "year"))
```

3.4 Clean Up Data

3.4.1 Remove unnecessary variables from `tuberculosis` .

```
tuberculosis <- tuberculosis %>%
  select(country, year, type, sex, age, cases, population)
```

3.4.2 Filter out rows with `NA` values.

3.4.3 Save the cleaned data back into `tuberculosis` .

```
tuberculosis <- tuberculosis %>%
  select(country, year, type, sex, age, cases, population) %>%
  drop_na()
```

4. Data Manipulation

4.1 Determine the total TB cases among men and women in the 21st century in the United States. Identify which sex had more cases.

```
us_cases <- tuberculosis %>%
  filter(
    country == "United States of America",
    year >= 2000
  ) %>%
  group_by(sex) %>%
  summarise(total_cases = sum(cases))

us_cases
```

```
## # A tibble: 0 × 2
## # ℹ 2 variables: sex <chr>, total_cases <dbl>
```

4.2 Create a new variable, `cases_per_100k` , representing TB cases per 100,000 people by year, sex, age group, and TB type.

```
tuberculosis <- tuberculosis %>%
  mutate(
    cases_per_100k = (cases / population) * 100000
  )
tuberculosis
```

```
## # A tibble: 61,282 × 8
##    country      year type  sex   age   cases population cases_per_100k
##    <chr>       <dbl> <chr> <chr> <chr> <dbl>      <dbl>          <dbl>
##  1 Afghanistan  1997 sp    m     014       0   18452091         0
##  2 Afghanistan  1997 sp    m     1524     10   18452091         0.0542
##  3 Afghanistan  1997 sp    m     2534      6   18452091         0.0325
##  4 Afghanistan  1997 sp    m     3544      3   18452091         0.0163
##  5 Afghanistan  1997 sp    m     4554      5   18452091         0.0271
##  6 Afghanistan  1997 sp    m     5564      2   18452091         0.0108
##  7 Afghanistan  1997 sp    m     65        0   18452091         0
##  8 Afghanistan  1997 sp    f     014       5   18452091         0.0271
##  9 Afghanistan  1997 sp    f     1524     38   18452091         0.206
## 10 Afghanistan  1997 sp    f     2534     36   18452091         0.195
## # ℹ 61,272 more rows
```

4.3 Identify:

- - The country and year with the highest cases per 100k.
  - The country and year with the lowest cases per 100k.

```
highest_cases <- tuberculosis %>%
  arrange(desc(cases_per_100k)) %>%
  slice(1)

highest_cases
```

```
## # A tibble: 1 × 8
##   country  year type  sex   age   cases population cases_per_100k
##   <chr>   <dbl> <chr> <chr> <chr> <dbl>      <dbl>          <dbl>
## 1 Samoa    2009 sp    f     65     1111     191513           580.
```

Lowest cases per 100k (non-zero to avoid misleading zeros)

```
lowest_cases <- tuberculosis %>%
  filter(cases_per_100k > 0) %>%
  arrange(cases_per_100k) %>%
  slice(1)

lowest_cases
```

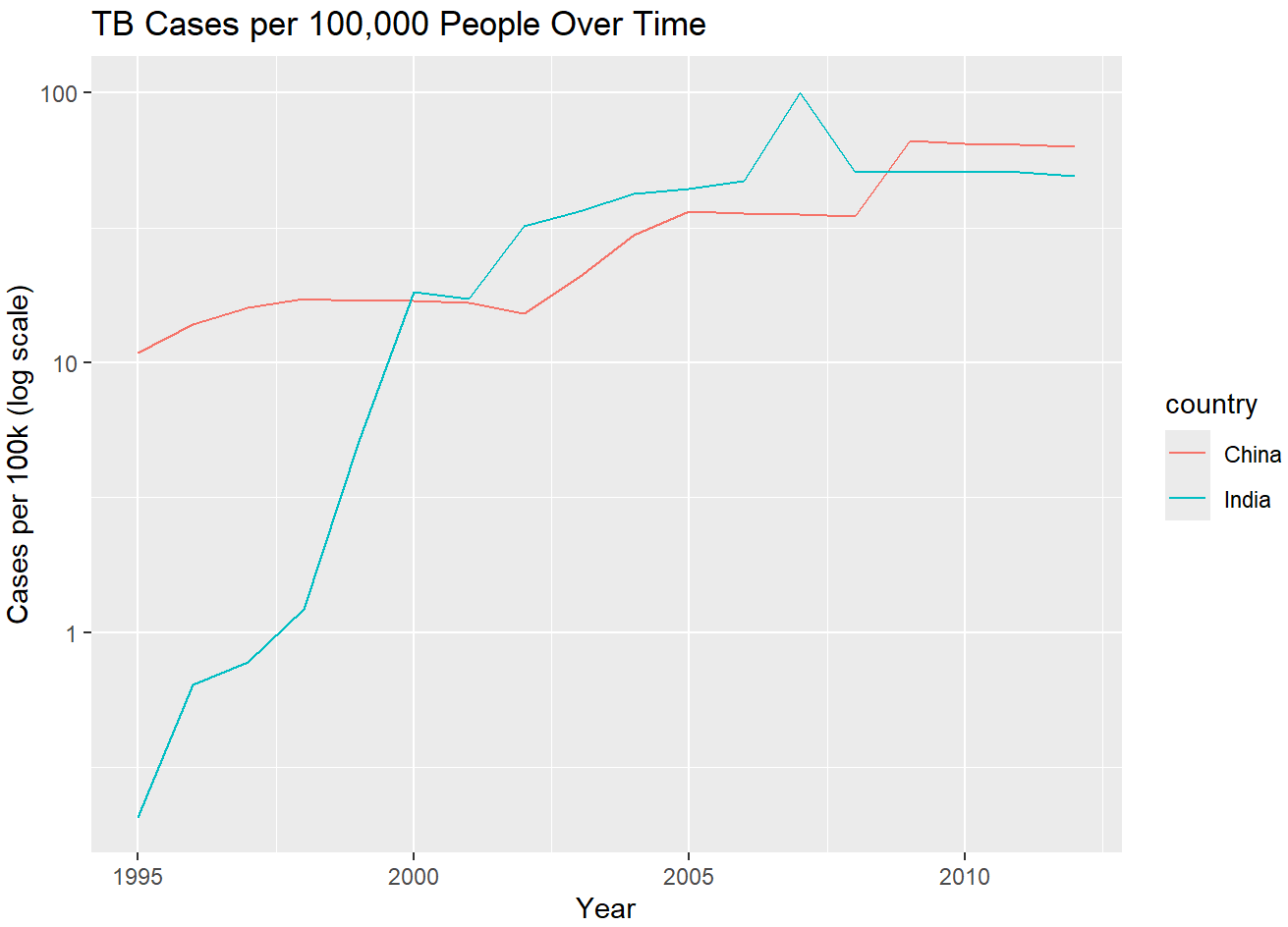
```
## # A tibble: 1 × 8
##   country             year type  sex   age   cases population cases_per_100k
##   <chr>              <dbl> <chr> <chr> <chr> <dbl>      <dbl>          <dbl>
## 1 Russian Federation  2000 sp    m     014       1  146596869       0.000682
```

5. Data Visualization

5.1 Plot the total cases per 100k as a function of year for China, India, and the United States:

- - Use a log scale on the y-axis (`scale_y_log10()`).
  - Describe emerging patterns.

```
tuberculosis %>%
  filter(country %in% c("China", "India", "United States of America")) %>%
  group_by(country, year) %>%
  summarise(total_cases_100k = sum(cases_per_100k), .groups = "drop") %>%
  ggplot(aes(x = year, y = total_cases_100k, color = country)) +
  geom_line() +
  scale_y_log10() +
  labs(
    title = "TB Cases per 100,000 People Over Time",
    y = "Cases per 100k (log scale)",
    x = "Year"
  )
```



5.2 Compare distributions of total cases per 100k (summed over years, sexes, and TB types) across age groups:

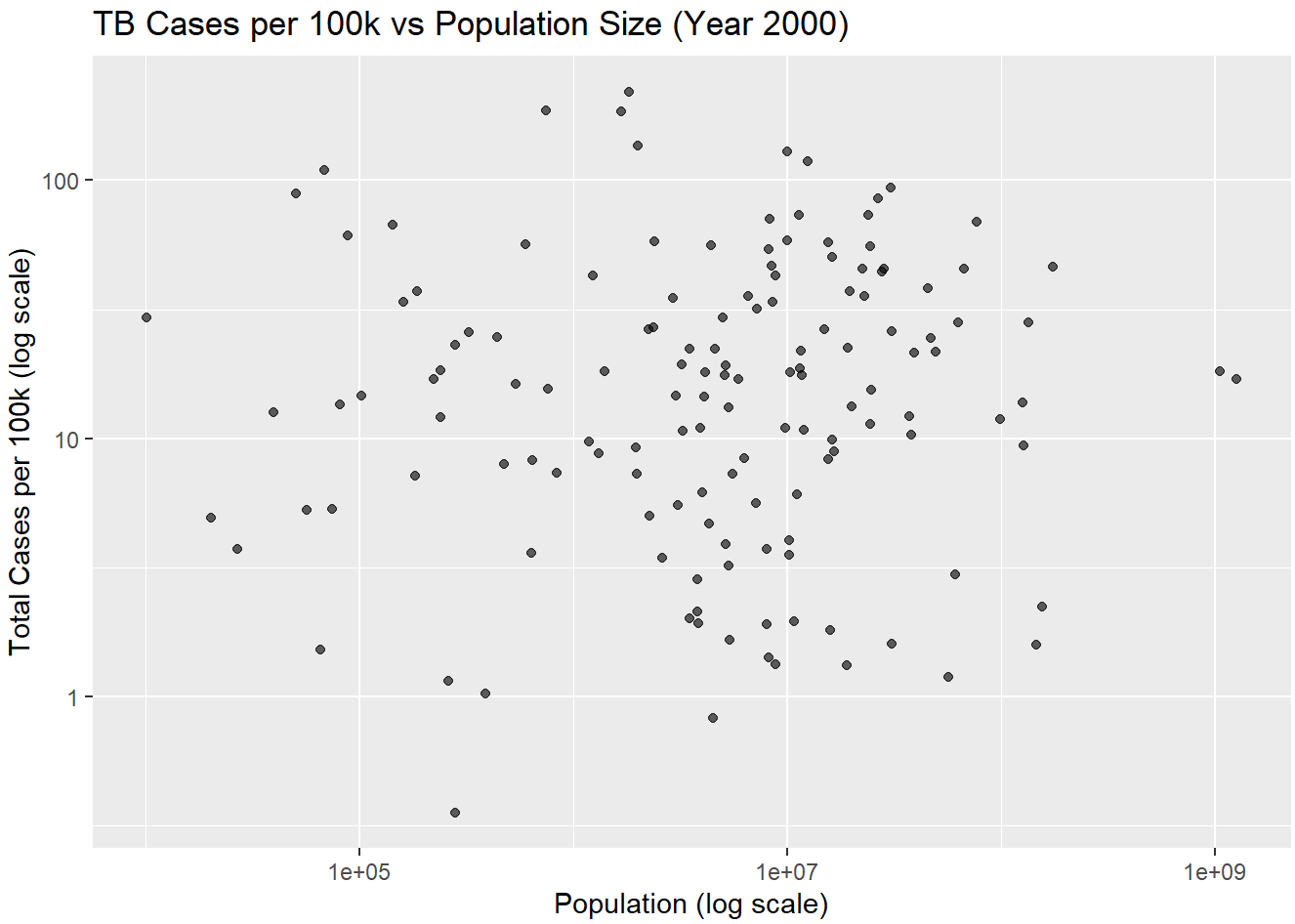
-
  - Use a log scale on the y-axis.
  - Highlight observed patterns.

```
tuberculosis %>%
  group_by(age) %>%
  summarise(total_cases_100k = sum(cases_per_100k), .groups = "drop") %>%
  ggplot(aes(x = age, y = total_cases_100k)) +
  geom_col() +
  scale_y_log10() +
  labs(
    title = "Distribution of TB Cases per 100k by Age Group",
    x = "Age Group",
    y = "Total Cases per 100k (log scale)"
  )
```

5.3 Create a plot to evaluate whether the number of cases per 100k in 2000 was related to a country's population:

- ◦ Conclude based on the visualization.

```
tuberculosis %>%
  filter(year == 2000) %>%
  group_by(country) %>%
  summarise(
    total_cases_100k = sum(cases_per_100k),
    population = mean(population),
    .groups = "drop"
  ) %>%
  ggplot(aes(x = population, y = total_cases_100k)) +
  geom_point(alpha = 0.6) +
  scale_x_log10() +
  scale_y_log10() +
  labs(
    title = "TB Cases per 100k vs Population Size (Year 2000)",
    x = "Population (log scale)",
    y = "Total Cases per 100k (log scale)"
  )
```



The visualization suggests no strong direct relationship between population size and TB cases per 100,000. Countries with large populations do not necessarily experience higher TB rates, indicating that TB burden is more closely linked to public health conditions than population size alone.