4.1 Clean `df_demographics`

(a) Some fields may have been missing or scarped incorrectly resulting in nonsensical values or `NaN` $\rightarrow$ we made sure columns have appropriate numeric data types, and dropped rows with missing values, or values not in expected range.

(b) All countries had valid, numeric life expectancy values, in range $[40, 100]$.

(c) All countries had valid, numeric population density and urban population values.

(d) Country names are inconsistent between datasets $\rightarrow$ In order to keep as many countries as possible when merging datasets down the line, we applied a normalization scheme to country names.

(f) Overall, 7 countries were affected. For Brunei, a leading space was removed. The rest of the mismatches were caused by irregular capitalization (as Python's `str.title()` does not handle words such as "and" or "of" and abbreviations).

Old value	New value
Antigua and Barbuda	Antigua And Barbuda
Bosnia and Herzegovina	Bosnia And Herzegovina
Brunei	Brunei
Côte d'Ivoire	Côte D'Ivoire
DR Congo	Dr Congo
State of Palestine	State Of Palestine
Trinidad and Tobago	Trinidad And Tobago

- Rows before cleaning: 200
- Rows after cleaning: 200 (0 dropped)

4.2 Clean `df_gdp`

Issues encountered:

- Non-numeric characters (commas, currency symbols, etc.) in the `GDP_per_capita_PPP` column → prevented direct float conversion
- Missing values introduced when malformed strings were coerced to `NaN` → needed to be documented and removed
- Potential outliers in the GDP distribution → important to flag for downstream analysis
- Duplicate country entries → required de-duplication logic
- Inconsistent country names → needed standardization

Actions taken:

1. **Type conversion**
   Cast `GDP_per_capita_PPP` to string, stripped all non-digit/decimal characters, then converted to numeric (no invalid rows found).
2. **Missing-value handling**
   Exported rows where `GDP_per_capita_PPP` was `NaN` to `output/dropped_gdp.csv`, then dropped them from the DataFrame (no rows were dropped).

3. **Outlier detection (Tukey method)**

   Calculated Q1, Q3, and IQR; flagged values outside [Q1 – 1.5·IQR, Q3 + 1.5·IQR] as outliers (found 6 outliers).

4. **Duplicate removal**

   Removed duplicate rows by `Country` (no duplicates were found).

5. **Country-name standardization**

   Applied the same normalization as `df_demographics`.

- Rows before cleaning: 213
- Rows after cleaning: 213 (0 dropped)

4.3 Clean `df_pop`

Issues encountered:

- Non-numeric characters (spaces, commas, text) in the `Population` column $\rightarrow$ prevented direct numeric conversion
- Missing values created when malformed strings were coerced to `NaN` $\rightarrow$ needed to be documented and removed
- Potential outliers in the population distribution (on a $\log_{10}$ scale) $\rightarrow$ important to flag for review
- Duplicate country entries $\rightarrow$ required de-duplication logic
- Inconsistent country names $\rightarrow$ needed standardization

Actions taken:

1. **Type conversion**

   Cast Population to string, removed all non-digit/decimal characters, then converted to numeric (no invalid rows found).

2. **Missing-value handling**

   Rows Population was `NaN` were dropped (no such rows found).

3. **Outlier detection (Tukey on log scale)**

   Computed $\log_{10}(\texttt{Population})$, then calculated Q1, Q3, and IQR; flagged values outside [Q1 − 1.5·IQR, Q3 + 1.5·IQR] as outliers (1 outlier found).

4. **Duplicate removal**

   Removed duplicate rows by `Country` (no duplicates were found).

5. **Country-name standardization**

   Applied the same normalization as `df_demographics`.

- Rows before cleaning: 260
- Rows after cleaning: 260 (0 dropped)