CITS2402 - Introduction to Data Science

Assignment - "Migration and Cultural Diversity: An Analytical Comparison Between Australia and New Zealand"

Declaration

This declaration should be completed and remain attached to the top of your submission.

I/we am/are aware of the University's policy on academic conduct and I declare that this assignment is entirely the work of the author(s) listed below and that suitable acknowledgement has been made for any sources of information used in preparing it. I have retained a copy for my own records.

- [redacted]

Table of Contents:

1. Introduction

Australia and New Zealand are home to the Aboriginal and Torres Strait Islander peoples and the Māori, respectively. European settlement introduced migration, which has since shifted to focus on skilled migration, creating increasingly multicultural societies.

This project aims to compare migration patterns and cultural diversity by examining three key demographic features from the most recent census data: foreign-born populations, languages spoken, and religious affiliations. By analyzing these features, we seek to understand how migration has contributed to the evolving cultural identities of Australia and New Zealand.

2. Research Question:

How has migration shaped cultural diversity in Australia and New Zealand, as seen through net migration, languages spoken at home, and religious affiliations? This investigation will explore how migration has impacted multiculturalism, providing insights into the evolving social and cultural dynamics of each country.

3. Data Science Lifecycle

1. **Data Collection**: The data for this analysis was collected from the 2021 and 2016 Australian Census provided by the Australian Bureau of Statistics (ABS) and the 2018 and 2013 New Zealand Census from Stats NZ. These datasets contain demographic information on foreign-born populations, languages spoken at home, and religious affiliations.

2. **Data Processing**: The raw data required several transformations to make it suitable for analysis. This involved aligning the data structures from both countries, ensuring comparable fields such as "Country of Birth," "Languages Spoken at Home," and "Religious Affiliations" were standardized.

3. **Data Cleaning**: During the cleaning process, missing data points were handled appropriately. Irrelevant columns were removed, and consistent categories were established across both datasets. Any inconsistencies, such as differences in country names or data formats, were resolved by standardization. Assumptions made were carefully documented.

4. **Exploratory Data Analysis (EDA)**: Initial analysis was conducted to explore key statistics, such as the proportion of foreign-born residents in both countries, top countries of origin, and the distribution of languages spoken at home. This phase helped uncover initial trends and patterns that would guide the more in-depth analysis.

5. **Data Visualization**: Comparative visualizations were created to effectively communicate the differences and similarities between Australia and New Zealand. Bar charts and pie charts were used to illustrate migration patterns, the most common languages spoken, and religious affiliations. These visualizations provide a clear and insightful comparison of cultural diversity in both countries.

6. **Conclusion**: The findings of the analysis are summarized and interpreted in relation to the research question. Key insights on migration trends and cultural diversity are highlighted, with a discussion on how migration has shaped the multicultural landscapes of Australia and New Zealand.

4. Data Collection

This project utilizes publicly available data from the Australian and New Zealand statistical bureaus to examine migration, language diversity, and religious affiliations. The datasets enable a comprehensive comparison of cultural diversity between the two countries.

A) **Overall Comparison** (2021/2018 Data)

Australia:

1. **Migration**:
   - **Dataset**: ABS Overseas Migration Statistics (2021)
   - **File**: `migrationau.csv`
   - **Description**: Data on net migration, arrivals, and departures in Australia for 2021.
   - **Source**: ABS Overseas Migration
2. **Religion**:
   - **Dataset**: ABS Cultural Diversity Census (2021)
   - **File**: `religionau.csv`
   - **Description**: Religious affiliations of Australia's population in 2021, reflecting diversity driven by migration.
   - **Source**: ABS Cultural Diversity Census
3. **Language**:
   - **Dataset**: ABS Cultural Diversity in Australia (2021)
   - **File**: `aulanguage.csv`
   - **Description**: Data on the top languages spoken at home, excluding English, reflecting Australia's linguistic diversity.
   - **Source**: ABS Cultural Diversity in Australia

New Zealand:

1. **Migration**:
   - **Dataset**: Stats NZ Migration Data (2018)
   - **File**: `migrationnz.csv`
   - **Description**: Migration statistics for New Zealand, including net migration rates and arrivals by country of origin.
   - **Source**: Stats NZ Migration Data

2. **Religion**:
   - **Dataset**: Stats NZ Census Ethnic Groups (2018)
   - **File**: `nzreligion.csv`
   - **Description**: Religious affiliations in New Zealand, including the diversity brought by migration.
   - **Source**: Stats NZ Census Ethnic Groups
3. **Language**:
   - **Dataset**: Stats NZ Census Ethnic Groups (2018)
   - **File**: `nzlanguage.csv`
   - **Description**: Data on the top languages spoken in New Zealand homes, excluding English.
   - **Source**: Stats NZ Census Ethnic Groups

B) **Variation Analysis by Year** (2016/2013/2018 Data)

Australia (2016 Data):

1. **Religion**:
   - **Dataset**: ABS General Community DataPack (2016)
   - **File**: `aus_religion_2016.csv`
   - **Description**: Data on religious affiliations in Australia in 2016.
   - **Source**: ABS General Community DataPack
2. **Language**:
   - **Dataset**: ABS Cultural Diversity in Australia (2016)
   - **Files**: `aus_language_2016_A.csv`, `aus_language_2016_B.csv`, `aus_language_2016_C.csv`, `aus_language_2016_D.csv`, `aus_language_2016_E.csv`
   - **Description**: Data on top languages spoken at home (excluding English) in 2016.
   - **Source**: ABS General Community DataPack

New Zealand (2013, 2018 Data):

1. **Religion**:
   - **Dataset**: Religious Affiliation Data (2013, 2018)
   - **File**: `nz_religion_2013_2018.csv`
   - **Description**: Religious affiliations in New Zealand for the years 2013 and 2018.
   - **Source**: Aotearoa Data Explorer
   - Filters applied: 'Total people - age group', 'Total - New Zealand by District Health Board', 'Total people - birthplace', '2013', '2018'; Rows excluded: 'Total people - with at least one religious affiliation', 'Object to answering', 'Total people stated', 'Not elsewhere included'
2. **Language**:
   - **Dataset**: Languages Spoken Data (2013, 2018)
   - **File**: `nz_language_2013_2018.csv`

- **Description**: Information on the languages spoken in New Zealand for the years 2013 and 2018.
- **Source**: Aotearoa Data Explorer
- Filters applied: 'Total - New Zealand by District Health Board', 'Total people - ethnic group', '2013', '2018'; Rows excluded: 'None (eg too young to talk)', 'Total people stated', 'Not elsewhere included'

All data files have been converted to CSV format for easier processing in Python, and the file names have been standardized for consistency across analysis.

5. Assumptions:

5.1 Australia:

Migration

1) **Migrant** is defined as anyone residing in the country for 12 months or more, measured over a 16-month period.

2) **Migrant departures** occur when Australian residents leave for 12 months or more, measured over a 16-month period.

- Overseas migration data is based on the recorded movements of travellers crossing Australia's international border, with their exact duration of stay assessed.
- Net Overseas Migration arrivals and departures apply to all travellers regardless of nationality, citizenship, or visa type, including New Zealand and Australian citizens. Excluded from these statistics are travellers staying less than 12 months, air and ship crew, transit passengers, pleasure cruise passengers, and foreign diplomatic personnel and their families.

Religion

- For the 2016 census, the 'No Religion option' became the first response category in the Religious Affliation question.

Languages Spoken at Home

- For the 2016 census, the question only allows for one answer (respondents were only allowed to select one language spoken at home).
- This implies that the data represents a primary or dominant language.

5.2 New Zealand:

Migration

1) **Migrant** is defined as anyone residing in the country for 12 months or more of the following 16 months in the country

2) **Migrant departures** occurs when residents leave for 12 months or more, measured over a 16-month period.

Religion

- **Religious affiliation**: the self-identified association of a person with a religion,denomination, or sub-denominational religious group.

- A person can affiliate with more than one religion. A person affiliating with more than one religion is counted once in each applicable group at the level of the classification that is being used.

Languages Spoken

- **Language spoken**: the language(s) a person can speak or use. This includes New Zealand Sign Language and other sign languages.

- A person can report speaking or using more than one language. A person who reports speaking more than one language is counted once in each applicable group at the level of the classification that is being used.

5.3 Assumptions about Data

- Census data is comprehensive and includes all demographic groups.

- Self-reported data on language, religion, and migration is accurate.

- Religious groups have been categorized broadly (e.g., combining various Christian denominations) to simplify interpretation, though this reduces the level of detail.

- Similar languages were grouped together (e.g., dialects under one language) to simplify interpretation, though this reduces the level of detail.

6. Data Processing

The data processing stage involves preparing the raw datasets for analysis by following these steps:

1. **Uploading Files**:
   - The CSV files collected from the Australian and New Zealand censuses are uploaded from the local machine into the environment for further analysis.
2. **Reading the Files**:
   - The datasets are read into Python using **pandas** (`pd.read_csv()`). This ensures that the data is structured in a tabular format, allowing for easy manipulation and exploration.
3. **Initial Exploration**:
   - The first 6 rows of each dataset are printed using `head()` to observe the structure and identify important columns. This step helps understand how the data is organized (e.g., column names, data types, missing values) and informs the next steps in the data cleaning process.

We begin by reading in the datasets for both the overall comparison (migration, language, and religion) and variation analysis (data from previous years). This will allow us to explore the trends in cultural diversity in both Australia and New Zealand.

```
# Provides DataFrames for structured data handling
import pandas as pd

# Creates informative graphics, particularly for statistical plots
import seaborn as sns

# Offers a MATLAB-like interface for creating various charts
import matplotlib.pyplot as plt

# Provides support for numerical operations, including arrays and math
functions
import numpy as np
```

In this step, the data files are uploaded from the local machine and read into pandas DataFrames. This allows for structured data handling, making it easier to perform analysis and visualization.

```
# Uploading the files

from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Langauage2016A.csv to Langauage2016A (2).csv
Saving Language2016B.csv to Language2016B (2).csv
Saving Language2016C.csv to Language2016C (2).csv
Saving Language2016D.csv to Language2016D (2).csv
Saving nzlanguage.csv to nzlanguage (2).csv
Saving aur_religion_2016.csv to aur_religion_2016 (2).csv
Saving nz_language_2013_2018.csv to nz_language_2013_2018 (2).csv
Saving nz_religion_2013_2018.csv to nz_religion_2013_2018 (2).csv
Saving aulanguage.csv to aulanguage (2).csv
Saving religionau.csv to religionau (2).csv
Saving migrationau.csv to migrationau (2).csv
Saving migrationnz.csv to migrationnz (2).csv
Saving nzreligion.csv to nzreligion (2).csv
```

6.1 Reading Overall Comparison Files (2021 for Australia and 2018 for New Zealand)

We will first read the data for migration, language, and religion from both countries for the most recent census years available.

```
# Reading the files

# migration
aus_migration = pd.read_csv('migrationau.csv')
nz_migration = pd.read_csv('migrationnz.csv')

# langauge
aus_langauge = pd.read_csv('aulanguage.csv')
nz_langauge = pd.read_csv('nzlanguage.csv')

# religion
aus_religion = pd.read_csv('religionau.csv')
nz_religion = pd.read_csv('nzreligion.csv')

# reading first 6 rows of both files for  migration
# to understand the structure of csv

print(aus_migration.head())
print(nz_migration.head())
```

```
                                                         Graph 1.1 -
Overseas migration - Australia - year ending(a)
NaN    Migrant arrivals ('000) Migrant departures ('000)
Net overseas migration(b) ('000)
Jun-13 482.09                  -251.76
230.33
Sep-13 484.31                  -263.10
221.21
Dec-13 478.68                  -270.31
208.38
Mar-14 472.63                  -270.44
202.19
   Category  Migrant arrivals  Migrant departures  Net migration
0  Dec-2001            114597               84332          30265
1  Jan-2002            118012               81053          36959
2  Feb-2002            119546               77522          42024
3  Mar-2002            122621               75833          46788
4  Apr-2002            124293               74785          49508
```

```
# reading first 6 rows for langauge
print(aus_langauge.head())
print(nz_langauge.head())
```

```
Top 5 most common languages other than English, 2021
NaN Language   Persons who used language at home (count) Proportion of
population (%)     Proportion with low English proficiency (%)(a)
1   Mandarin   685,274                                   2.7
25.9
2   Arabic     367,159                                   1.4
15.3
```

```
3   Vietnamese 320,758                                  1.3
30.5
4   Cantonese  295,281                                  1.2
23.7
   Year  EthnicLevel EthnicValue Ethnic_group_description  \
0  2018            4       12934                    Gypsy
1  2018            4       51120                 Lebanese
2  2018            4       12914                  Belgian
3  2018            4       42116                Taiwanese
4  2018            4       12116                    Irish

  Languages_spoken_code Languages_spoken_description  \
0                 13110                          Yue
1                 04213                    Bulgarian
2                 01212                      Swedish
3                 11100   Uralic not further defined
4                 01113                       German

   Census_usually_resident_population_count Percentage
0                                         0        0.0
1                                         0        0.0
2                                         0        0.0
3                                         0        0.0
4                                       231        1.3
```

```python
#reading first 6 rows for religion
print(aus_religion.head())
print(nz_religion.head())
```

```
                     Australian Bureau of Statistics  \
0  Census of Population and Housing: Census artic...
1    Released at 10:00am (Canberra time) 4 July 2022
2                                                NaN
3  TABLE 6. RELIGIOUS AFFLIATION (NARROW GROUPS) ...
4                              IN AUSTRALIA - 2021

                     Unnamed: 1  Unnamed: 2  Unnamed: 3  Unnamed: 4  \
0                           NaN         NaN         NaN         NaN
1                      Contents         NaN         NaN         NaN
2                 Find out more:        NaN         NaN         NaN
3          Religious affiliation        NaN         NaN         NaN
4  Year of arrival in Australia         NaN         NaN         NaN

   Unnamed: 5  Unnamed: 6  Unnamed: 7  Unnamed: 8  Unnamed: 9
Unnamed: 10  \
0         NaN         NaN         NaN         NaN         NaN
NaN
1         NaN         NaN         NaN         NaN         NaN
NaN
```

```
2         NaN         NaN         NaN         NaN         NaN
NaN
3         NaN         NaN         NaN         NaN         NaN
NaN
4         NaN         NaN         NaN         NaN         NaN
NaN

   Unnamed: 11  Unnamed: 12  Unnamed: 13  Unnamed: 14  Unnamed: 15  \
0          NaN          NaN          NaN          NaN          NaN
1          NaN          NaN          NaN          NaN          NaN
2          NaN          NaN          NaN          NaN          NaN
3          NaN          NaN          NaN          NaN          NaN
4          NaN          NaN          NaN          NaN          NaN

   Unnamed: 16  Unnamed: 17  Unnamed: 18  Unnamed: 19
0          NaN          NaN          NaN          NaN
1          NaN          NaN          NaN          NaN
2          NaN          NaN          NaN          NaN
3          NaN          NaN          NaN          NaN
4          NaN          NaN          NaN          NaN
   Year  EthnicLevel EthnicValue       Ethnic_group_description  \
0  2006            4       10000  European, not further defined
1  2006            4       10000  European, not further defined
2  2006            4       10000  European, not further defined
3  2006            4       10000  European, not further defined
4  2006            4       10000  European, not further defined

  Religious_affiliation_code Religious_affiliation_description  \
0                         00                       No religion
1                         01                          Buddhism
2                         02                         Christian
3                         03                          Hinduism
4                         04                             Islam

  Census_usually_resident_population_count Percentage
0                                     6903       32.8
1                                      159        0.8
2                                    12147       57.8
3                                       57        0.3
4                                      111        0.5
```

6.2 Reading Variation Files (2016 for Australia, 2013/2018 for New Zealand)

Next, we will load the datasets from earlier census years to analyze trends and variations in language and religion over time.

```
# Step 2: Reading the Variation Files
```

```python
# Australia 2016 data
religion_au_2016 = pd.read_csv('aur_religion_2016.csv')       #
Religion data (2016)

languageA = pd.read_csv('Langauage2016A.csv')       # Language data
part A (2016)
languageB = pd.read_csv('Language2016B.csv')        # Language data
part B (2016)
languageC = pd.read_csv('Language2016C.csv')            # Language
data part C (2016)
languageD= pd.read_csv('Language2016D.csv')        # Language data
part D (2016)

# New Zealand 2013/2018 data
religion_nz_2013_2018 = pd.read_csv('nz_religion_2013_2018.csv')  #
Religion data (2013/2018)
language_nz_2013_2018 = pd.read_csv('nz_language_2013_2018.csv')  #
Language data (2013/2018)


# Reading first 6 rows of variation files to understand the structure

# Australia 2016 religion and language variation files

print(religion_au_2016.head())
print(languageA.head())
print(languageB.head())
print(languageC.head())
print(languageD.head())


# New Zealand 2013/2018 religion and language variation files

print(religion_nz_2013_2018.head())
print(language_nz_2013_2018.head())
```

```
  AUS_CODE_2021  Buddhism_M  Buddhism_F  Buddhism_P
Christianity_Anglican_M  \
0           AUS      265305      350514      615823
1135624

   Christianity_Anglican_F  Christianity_Anglican_P  \
0                  1360653                  2496273

   Christianity_Asyrin_Apstlic_M  Christianity_Asyrin_Apstlic_F  \
0                           9253                           9880

   Christianity_Asyrin_Apstlic_P  ...  SB_OSB_NRA_OSB_P
SB_OSB_NRA_Tot_M  \
0                          19141  ...             45970
5122953
```

```
   SB_OSB_NRA_Tot_F  SB_OSB_NRA_Tot_P  Religious_affiliation_ns_M  \
0           4764004           9886957                      984981

   Religious_affiliation_ns_F  Religious_affiliation_ns_P     Tot_M
Tot_F  \
0                      863446                     1848426  12545154
12877635

      Tot_P
0  25422788

[1 rows x 103 columns]
   AUS_CODE_2016  MSEO_SEO MSEO_SOLSE_VWorW MSEO_SOLSE_NWorNAA
MSEO_SOLSE_Tot  \
0             36   8417802               ..                 ..
..

  MSEO_NS  MSEO_Tot MOL_Afrikaans_SEO  MOL_Afrikaans_SOLSE_VWorW  \
0      ..   8417802                ..                      21164

   MOL_Afrikaans_SOLSE_NWorNAA  ...  MOL_SAL_Indon_NS
MOL_SAL_Indon_Tot  \
0                          338  ...               230
30894

   MOL_SAL_Tagalog_SEO MOL_SAL_Tagalog_SOLSE_VWorW  \
0                   ..                       43994

   MOL_SAL_Tagalog_SOLSE_NWorNAA  MOL_SAL_Tagalog_SOLSE_Tot  \
0                           1333                      45333

   MOL_SAL_Tagalog_NS  MOL_SAL_Tagalog_Tot  MOL_SAL_Oth_SEO  \
0                 410                45736               ..

  MOL_SAL_Oth_SOLSE_VWorW
0                   11561

[1 rows x 201 columns]
   AUS_CODE_2016  MOL_SAL_Oth_SOLSE_NWorNAA  MOL_SAL_Oth_SOLSE_Tot  \
0             36                        924                  12482

   MOL_SAL_Oth_NS  MOL_SAL_Oth_Tot MOL_SAL_Tot_SEO
MOL_SAL_Tot_SOLSE_VWorW  \
0             111            12598              ..
112111

   MOL_SAL_Tot_SOLSE_NWorNAA  MOL_SAL_Tot_SOLSE_Tot
MOL_SAL_Tot_NS  ...  \
0                       5557                 117673
```

```
986  ...

   FOL_Japanese_SEO FOL_Japanese_SOLSE_VWorW
FOL_Japanese_SOLSE_NWorNAA  \
0                ..                    27607
5083

   FOL_Japanese_SOLSE_Tot  FOL_Japanese_NS  FOL_Japanese_Tot
FOL_Korean_SEO  \
0                   32686              249             32939
..

  FOL_Korean_SOLSE_VWorW  FOL_Korean_SOLSE_NWorNAA
FOL_Korean_SOLSE_Tot
0                  37660                     18568
56225

[1 rows x 201 columns]
   AUS_CODE_2016  FOL_Korean_NS  FOL_Korean_Tot FOL_Macedonian_SEO  \
0             36            459           56687                 ..

   FOL_Macedonian_SOLSE_VWorW  FOL_Macedonian_SOLSE_NWorNAA  \
0                       26930                          6175

   FOL_Macedonian_SOLSE_Tot  FOL_Macedonian_NS  FOL_Macedonian_Tot  \
0                     33106                343               33449

  FOL_Maltese_SEO  ...  POL_French_SOLSE_NWorNAA  POL_French_SOLSE_Tot
\
0              ..  ...                      2983                 70206


   POL_French_NS  POL_French_Tot  POL_German_SEO
POL_German_SOLSE_VWorW  \
0            668           70873              ..
76191

   POL_German_SOLSE_NWorNAA  POL_German_SOLSE_Tot  POL_German_NS  \
0                      2262                 78456            900

   POL_German_Tot
0           79353

[1 rows x 201 columns]
   AUS_CODE_2016 POL_Greek_SEO  POL_Greek_SOLSE_VWorW  \
0             36            ..                 197651

   POL_Greek_SOLSE_NWorNAA  POL_Greek_SOLSE_Tot  POL_Greek_NS
POL_Greek_Tot  \
0                    37619               235267          2321
```

```
237588

  POL_IAL_Bengali_SEO  POL_IAL_Bengali_SOLSE_VWorW  \
0                   ..                        50019

   POL_IAL_Bengali_SOLSE_NWorNAA  ...  P_LSatH_NS_SOLSE_NWorNAA  \
0                           4199  ...                      9927

   P_LSatH_NS_SOLSE_Tot  P_LSatH_NS_NS P_LSatH_NS_Tot  P_Tot_SEO  \
0                 69335        1440493        1509829   17020421

   P_Tot_SOLSE_VWorW  P_Tot_SOLSE_NWorNAA  P_Tot_SOLSE_Tot  P_Tot_NS
P_Tot_Tot
0            4068598               819925          4888523   1492943
23401892

[1 rows x 193 columns]
  STRUCTURE                STRUCTURE_ID  \
0  DATAFLOW  STATSNZ:CEN18_ECI_026(1.0)
1  DATAFLOW  STATSNZ:CEN18_ECI_026(1.0)
2  DATAFLOW  STATSNZ:CEN18_ECI_026(1.0)
3  DATAFLOW  STATSNZ:CEN18_ECI_026(1.0)
4  DATAFLOW  STATSNZ:CEN18_ECI_026(1.0)

                                      STRUCTURE_NAME ACTION  \
0  Religious affiliation (total responses) and bi...      I
1  Religious affiliation (total responses) and bi...      I
2  Religious affiliation (total responses) and bi...      I
3  Religious affiliation (total responses) and bi...      I
4  Religious affiliation (total responses) and bi...      I

   AGE_CEN18_ECI_026                Age group AREA_CEN18_ECI_026  \
0             999999  Total people - age group            DHB9999
1             999999  Total people - age group            DHB9999
2             999999  Total people - age group            DHB9999
3             999999  Total people - age group            DHB9999
4             999999  Total people - age group            DHB9999

                                           Area
BIRTHPLACE_CEN18_ECI_026  \
0  Total - New Zealand by District Health Board
99
1  Total - New Zealand by District Health Board
99
2  Total - New Zealand by District Health Board
99
3  Total - New Zealand by District Health Board
99
4  Total - New Zealand by District Health Board
99
```

```
                  Birthplace  RELIGION_CEN18_ECI_026 Religious
affiliation  \
0  Total people - birthplace                      16
Buddhism
1  Total people - birthplace                      16
Buddhism
2  Total people - birthplace                      17
Hinduism
3  Total people - birthplace                      17
Hinduism
4  Total people - birthplace                      18
Islam

   YEAR_CEN18_ECI_026  Year  OBS_VALUE  Observation value  OBS_STATUS
\
0                2013   NaN    58407.0                NaN         NaN

1                2018   NaN    52761.0                NaN         NaN

2                2013   NaN    89916.0                NaN         NaN

3                2018   NaN   123384.0                NaN         NaN

4                2013   NaN    46146.0                NaN         NaN


   Observation status
0                 NaN
1                 NaN
2                 NaN
3                 NaN
4                 NaN
  STRUCTURE                STRUCTURE_ID  \
0  DATAFLOW  STATSNZ:CEN18_ECI_006(1.0)
1  DATAFLOW  STATSNZ:CEN18_ECI_006(1.0)
2  DATAFLOW  STATSNZ:CEN18_ECI_006(1.0)
3  DATAFLOW  STATSNZ:CEN18_ECI_006(1.0)
4  DATAFLOW  STATSNZ:CEN18_ECI_006(1.0)

                                      STRUCTURE_NAME ACTION  \
0  Birthplace (New Zealand or overseas) and ethni...      I
1  Birthplace (New Zealand or overseas) and ethni...      I
2  Birthplace (New Zealand or overseas) and ethni...      I
3  Birthplace (New Zealand or overseas) and ethni...      I
4  Birthplace (New Zealand or overseas) and ethni...      I

  AREA_CEN18_ECI_006                                         Area  \
0            DHB9999  Total - New Zealand by District Health Board
1            DHB9999  Total - New Zealand by District Health Board
```

```
2           DHB9999  Total - New Zealand by District Health Board
3           DHB9999  Total - New Zealand by District Health Board
4           DHB9999  Total - New Zealand by District Health Board

   BIRTHPLACE_CEN18_ECI_006                 Birthplace
ETHNIC_CEN18_ECI_006  \
0                        99  Total people - birthplace
9999
1                        99  Total people - birthplace
9999
2                        99  Total people - birthplace
9999
3                        99  Total people - birthplace
9999
4                        99  Total people - birthplace
9999

                  Ethnic group  LANGUAGE_CEN18_ECI_006 Languages
spoken  \
0  Total people - ethnic group                       1
English
1  Total people - ethnic group                       1
English
2  Total people - ethnic group                       2
Maori
3  Total people - ethnic group                       2
Maori
4  Total people - ethnic group                       3
Samoan

   YEAR_CEN18_ECI_006  Year  OBS_VALUE  Observation value  OBS_STATUS
\
0                2013   NaN  3819972.0                NaN         NaN

1                2018   NaN  4482132.0                NaN         NaN

2                2013   NaN   148395.0                NaN         NaN

3                2018   NaN   185955.0                NaN         NaN

4                2013   NaN    86403.0                NaN         NaN


   Observation status
0                NaN
1                NaN
2                NaN
3                NaN
4                NaN
```

7. Data Cleaning

The data cleaning process involves identifying and handling any missing values, correcting data types, renaming columns if necessary, and filtering out irrelevant data. This ensures that the datasets are ready for analysis.

7.1 Cleaning Overall Comparison Files (2021 for Australia, 2018 for New Zealand)

We will start by cleaning the migration, language, and religion datasets for both countries to ensure consistency and accuracy. This involves handling missing values, renaming columns for clarity, and ensuring correct data types.

We will clean the datasets in three stages: migration, language, and religion.

1. Migration Data Cleaning

During the cleaning process for the migration data, several issues were encountered and resolved as follows:

Issues

1. Australia's data loaded as one column instead of many
2. Dates were in different formats between countries
3. Both datasets had extra rows with notes and metadata
4. Australia had negative numbers for people leaving
5. Australian data needed scaling

Solutions

1. Used special settings when reading Australia's file to split columns correctly
2. Changed dates in both datasets to a standard format
3. Kept only rows with actual numbers, removing notes and metadata
4. Removed negative numbers (minus sign) for departures in the Australian dataset using abs()
5. Multiplied the numeric columns by 1000 for Australia

```
# column check on australian migration file
print(aus_migration.columns)
```

```
Index(['Graph 1.1 - Overseas migration - Australia - year ending(a)'],
dtype='object')
```

```
# Function to clean and process Australian migration data
def read_data_aus_migration(file_path):
    data = []  # Initialize list for cleaned data


    # Open file and read lines
    with open(file_path, 'r') as file:
        for i, line in enumerate(file):
```

```
            if i == 0:
                continue  # Skip header
            cleaned_line = line.strip().split(',')  # Split line by
comma
            cleaned_line = [entry.replace('"', '') for entry in
cleaned_line]  # Remove quotes
            data.append(cleaned_line)  # Add cleaned line to list


    # Filter rows with valid numeric data
    filtered_data = [row for row in data if len(row) >= 4 and
row[1].replace('.', '', 1).isdigit()]


    # Convert numeric columns to abs values and scale by 1000
    for row in filtered_data:
        # Convert and scale 'Migrant Arrivals' as is
        row[1] = str(int(float(row[1]) * 1000))

        # Convert and scale 'Migrant Departures' to absolute value
        row[2] = str(int(abs(float(row[2])) * 1000))


        # Convert and scale 'Net Migration' as is
        row[3] = str(int(float(row[3]) * 1000))


    # Create DataFrame from filtered data
    df = pd.DataFrame(filtered_data, columns=['Date', 'Migrant
Arrivals', 'Migrant Departures', 'Net Migration'])


    # Convert 'Date' to datetime and format
    df['Date'] = pd.to_datetime(df['Date'], format='%b-
%y').dt.strftime('%b-%Y')


    # Convert numeric columns to integers
    for col in ['Migrant Arrivals', 'Migrant Departures', 'Net
Migration']:
        df[col] = pd.to_numeric(df[col], downcast='integer')


    return df


# Read and clean the migration data
data_aus_migration = 'migrationau.csv'
aus_migration_cleaned = read_data_aus_migration(data_aus_migration)
```

```python
# Show cleaned data
print(aus_migration_cleaned.head())
```

```
       Date  Migrant Arrivals  Migrant Departures  Net Migration
0  Jun-2013            482090              251760         230330
1  Sep-2013            484310              263100         221210
2  Dec-2013            478680              270310         208380
3  Mar-2014            472630              270440         202190
4  Jun-2014            464680              276900         187780
```

```python
# Function to clean and process New Zealand migration data
def read_data_nz_migration(file_path):
    data = []  # Initialize list for cleaned data

    # Open file and read lines
    with open(file_path, 'r') as file:
        for line in file:
            cleaned_line = line.strip().split(',')  # Split by comma
            cleaned_line = [entry.replace('"', '') for entry in
cleaned_line]  # Remove quotes
            data.append(cleaned_line)  # Add cleaned line to list

    # Filter rows with valid numeric data
    filtered_data = [row for row in data if len(row) >= 4 and
row[1].replace('.', '', 1).isdigit()]

    # Create DataFrame from filtered data
    df = pd.DataFrame(filtered_data, columns=['Date', 'Migrant
Arrivals', 'Migrant Departures', 'Net Migration'])

    # Convert 'Date' to datetime and format
    df['Date'] = pd.to_datetime(df['Date'], format='%b-%Y')
    df['Date'] = df['Date'].dt.strftime('%b-%Y')

    # Convert numeric columns to numbers, handling any errors
    for col in ['Migrant Arrivals', 'Migrant Departures', 'Net
Migration']:
        df[col] = pd.to_numeric(df[col], errors='coerce')

    return df  # Return cleaned DataFrame

# Read and clean New Zealand migration data
data_nz_migration = 'migrationnz.csv'
nz_migration_cleaned = read_data_nz_migration(data_nz_migration)

# Show cleaned data
print(nz_migration_cleaned.head())
```

```
       Date  Migrant Arrivals  Migrant Departures  Net Migration
0  Dec-2001            114597               84332          30265
1  Jan-2002            118012               81053          36959
```

```
2   Feb-2002              119546             77522         42024
3   Mar-2002              122621             75833         46788
4   Apr-2002              124293             74785         49508
```

2. Language Data Cleaning

Issues

1. Unwanted rows, categories, and columns (e.g., "English," "Total," metadata, unnecessary columns).
2. Data formatting issues (commas in numeric columns, data type problems).
3. Duplicate language entries.
4. Inconsistent column names.
5. Australia's dataset already provided the top 5 languages excluding English, so we made New Zealand's data consistent by selecting the top 5 languages, excluding English, as well.

Solutions

1. **Filtered out** unwanted categories and kept only relevant columns ('Language' and 'Count').
2. **Removed commas** from numeric data and converted to appropriate data types (integer).
3. **Grouped by language** and summed population counts for duplicate entries.
4. **Renamed columns** to 'Language' and 'Count' for consistency between datasets.
5. **Sorted by count** and selected the top 5 languages for New Zealand, excluding English, to match Australia's top 5 language data.

```python
# Function to clean Australian language data
def clean_aus_language_data(file_path):
    # Read the CSV file, skipping the first row
    df = pd.read_csv(file_path, skiprows=1)

    # Drop unnecessary columns
    df.drop(columns=['Unnamed: 0'], inplace=True)

    # Keep only the 'Language' and 'Persons who used language at home 
(count)' columns
    df = df[['Language', 'Persons who used language at home (count)']]

    # Remove extra spaces from language names
    df['Language'] = df['Language'].str.strip()

    # Remove commas from numeric values and convert to float
    df['Persons who used language at home (count)'] = df['Persons who 
used language at home (count)'].str.replace(',', '').astype(float)

    # Drop rows with missing values
    df.dropna(subset=['Language', 'Persons who used language at home 
(count)'], inplace=True)
```

```
    # Convert the count to integer type
    df['Persons who used language at home (count)'] = df['Persons who
used language at home (count)'].astype(int)

    # Rename the columns to 'Language' and 'Count'
    df.columns = ['Language', 'Count']

    return df

# File path for the Australian language data
file_path_aus = 'aulanguage.csv'

# Clean the Australian language data
aus_language_cleaned = clean_aus_language_data(file_path_aus)

# Display the cleaned data
print(aus_language_cleaned)
```

```
     Language   Count
0    Mandarin  685274
1      Arabic  367159
2  Vietnamese  320758
3   Cantonese  295281
4     Punjabi  239033
```

```
# Function to clean New Zealand language data and get top 5 languages
def clean_nz_language_data(file_path):
    # Read the CSV file
    df = pd.read_csv(file_path)

    # Remove unwanted language categories
    unwanted_values = ['English', 'Total', 'Total stated', 'None (eg
too young to talk)',
                       'Don\'t know', 'Refused to answer', 'Response
unidentifiable', 'Not stated']
    df = df[~df['Languages_spoken_description'].isin(unwanted_values)]

    # Group by language and sum population counts
    language_counts = df.groupby('Languages_spoken_description')
['Census_usually_resident_population_count'].sum()

    # Reset index and rename columns
    language_counts_df = language_counts.reset_index()
    language_counts_df.columns = ['Language', 'Count']

    # Sort by count and select top 5 languages
    top_5_languages = language_counts_df.sort_values(by='Count',
ascending=False).head(5)

    # Reset index to get ranking numbers starting from 1
    top_5_languages.reset_index(drop=True, inplace=True)
```

```python
    return top_5_languages

# File path for New Zealand language data
file_path_nz = 'nzlanguage.csv'

# Get top 5 languages from the New Zealand data
nz_top5_languages = clean_nz_language_data(file_path_nz)

# Display the top 5 languages with proper numbering
nz_top5_languages.index = nz_top5_languages.index + 1
print(nz_top5_languages)
```

```
           Language   Count
1             Māori  371277
2            Samoan  330798
3  Northern Chinese  293424
4             Hindi  273201
5            French  250053
```

Religion Data Cleaning

Problem:

1. The dataset contains irrelevant metadata rows and empty columns.
2. The `Count` column has values formatted as strings with commas.
3. Some rows, like "Total," are not relevant to the analysis of individual religions.
4. Population counts are being displayed with decimal points, but they should be integers.
5. The datasets for Australia and New Zealand were in different formats, making direct comparison difficult.

Solution:

1. **Dropping** empty rows and columns to remove metadata.
2. **Converting** the `Count` column into numeric format by removing commas and converting it to integers.
3. **Filtering out** rows containing non-religion labels like "Total" to focus on relevant religious categories.
4. **Sorting** the dataset by population count and extracting the top 7 religions with the highest counts.
5. **Standardizing** the formats of the Australian and New Zealand datasets to ensure consistency, simplifying the comparison of the top 7 religions between both countries.

Explanation of Initial Datasets

Australian Dataset:

- The initial dataset contains metadata rows and many `Unnamed` columns that are irrelevant to the analysis.

- The relevant columns (religions and counts) are scattered, requiring you to extract only the essential data for analysis.

New Zealand Dataset:

- The New Zealand dataset is more structured, but it contains irrelevant rows like "Total," which need to be removed.
- The population counts are stored as strings with commas, and these must be converted into numeric values for consistent analysis.

```
# Load the Australian religion dataset
aus_religion = pd.read_csv('religionau.csv')

# Drop the first 6 rows (metadata)
aus_religion_cleaned = aus_religion.drop([0, 1, 2, 3, 4,
5]).reset_index(drop=True)

# Remove completely empty columns
aus_religion_cleaned = aus_religion_cleaned.dropna(axis=1, how='all')

# Keep only 'Religion' and 'Count' columns
aus_religion_cleaned = aus_religion_cleaned.iloc[:, [0, 1]]

# Rename columns for consistency
aus_religion_cleaned.columns = ['Religion', 'Count']

# Drop rows with missing values
aus_religion_cleaned = aus_religion_cleaned.dropna(subset=['Religion',
'Count'])

# Remove commas from 'Count' and convert to integer
aus_religion_cleaned['Count'] =
aus_religion_cleaned['Count'].str.replace(',', '').astype(int)

# Remove rows containing 'Total'
aus_religion_cleaned =
aus_religion_cleaned[~aus_religion_cleaned['Religion'].str.contains('T
otal', case=False)]

# Sort by 'Count' in descending order
aus_religion_sorted = aus_religion_cleaned.sort_values(by='Count',
ascending=False)

# Select top 7 religions
top_7_aus_religions =
aus_religion_sorted.head(7).reset_index(drop=True)

# Sum the counts of religions not in the top 7
others_count = aus_religion_sorted[7:]['Count'].sum()

# Create a row for 'Others'
others_row = pd.DataFrame({'Religion': ['Others'], 'Count':
```

```
[others_count]})

# Append 'Others' to the top 7 religions
top_8_aus_religions = pd.concat([top_7_aus_religions, others_row],
ignore_index=True)

# Display the final DataFrame
print(top_8_aus_religions)
```

```
                     Religion   Count
0  No Religion, (so described)  288211
1                     Hinduism  175873
2                     Catholic  151581
3                        Islam  100877
4                     Buddhism   77764
5                      Sikhism   47759
6            Christianity, nfd   39168
7                       Others  138776
```

```
# Load the New Zealand religion dataset
nz_religion = pd.read_csv('nzreligion.csv')

# Keep only 'Religion' and 'Count' columns
nz_religion_cleaned =
nz_religion[['Religious_affiliation_description',
'Census_usually_resident_population_count']]

# Rename columns for consistency
nz_religion_cleaned.columns = ['Religion', 'Count']

# Use .loc[] to remove commas from 'Count' and convert to numeric
nz_religion_cleaned.loc[:, 'Count'] =
pd.to_numeric(nz_religion_cleaned['Count'].str.replace(',', ''),
errors='coerce')

# Drop rows with missing values
nz_religion_cleaned = nz_religion_cleaned.dropna(subset=['Religion',
'Count'])

# Filter out irrelevant terms like 'Total', 'Object to answering',
etc.
irrelevant_terms = ['Total', 'Object to answering', 'Not elsewhere
included', 'Total stated']
nz_religion_cleaned =
nz_religion_cleaned[~nz_religion_cleaned['Religion'].str.contains('|'.
join(irrelevant_terms), case=False)]

# Group by 'Religion' and sum the counts (in case of duplicates)
nz_religion_cleaned = nz_religion_cleaned.groupby('Religion',
as_index=False)['Count'].sum()
```

```
# Convert 'Count' to integer
nz_religion_cleaned['Count'] =
nz_religion_cleaned['Count'].astype(int)

# Sort by 'Count' in descending order
nz_religion_sorted = nz_religion_cleaned.sort_values(by='Count',
ascending=False)

# Select the top 7 religions
top_7_nz_religions = nz_religion_cleaned.sort_values(by='Count',
ascending=False).head(7).reset_index(drop=True)

# Calculate the sum of counts for religions not in the top 7
others_count = nz_religion_sorted[7:]['Count'].sum()

# Create a DataFrame for the 'Others' row
others_row = pd.DataFrame({'Religion': ['Others'], 'Count':
[others_count]})

# Append the 'Others' row to the top 7 religions DataFrame
top_8_nz_religions = pd.concat([top_7_nz_religions, others_row],
ignore_index=True)

# Display the final DataFrame with top 7 religions + 'Others'
print(top_8_nz_religions)
```

```
                                   Religion     Count
0                                 Christian  69678897
1                               No religion  65434410
2                                  Hinduism   3363279
3  Māori religions, beliefs and philosophies   2384751
4                                  Buddhism   2003550
5  Other religions, beliefs and philosophies   1792656
6                                     Islam   1749216
7                                    Others    968610
```

7.2 Cleaning Variation Files (2016 for Australia, 2013/2018 for New Zealand)

Next, we clean the variation files, ensuring they are consistent with the overall files. This will include merging language parts, if necessary, and standardizing column names across all datasets.

7.2.1 Language Data Cleaning (Australia 2016)

The 2016 Australian language data was split across five files. The following steps were applied to clean and consolidate the data:

Issues:

1. **Multiple Files**: Data was spread across five files.

2. **Irrelevant Columns**: Many columns were unnecessary.
3. **Data Orientation:** The data needed transposing for easier aggregation.
4. **Non-Numeric Values:** Some columns contained non-numeric data.
5. **Top Languages**: The top 25 languages needed to be identified.
6. **Handling "Others"**: Remaining languages needed to be grouped into an "Others" category.

Solutions:

1. **Concatenation**: Merged the five files using `pd.concat()`.
2. **Column Filtering**: Retained only columns ending with `'Tot'` (excluding those with "SOLSE").
3. **Transposing**: Transposed the DataFrame for row-wise calculations.
4. **Numeric Conversion**: Converted non-numeric values to `NaN` for proper summation.
5. **Top 25 Languages**: Calculated and selected the top 25 languages by total speakers.
6. **"Others" Category**: Aggregated remaining languages into an "Others" category.

The final DataFrame contains the top 25 languages and an "Others" category for remaining languages.

```
# Combine DataFrames
combined_df = pd.concat([languageA, languageB, languageC, languageD])

# Filter columns ending with 'Tot' but not containing 'SOLSE'
def column_filter(col_name):
    return col_name.endswith('Tot') and 'SOLSE' not in col_name

filtered_df = combined_df.loc[:, [col for col in combined_df.columns
if column_filter(col)]].copy()

# Transpose and sum rows
rotated_df = filtered_df.transpose()
rotated_df['Total'] = rotated_df.sum(axis=1)
combined_df = rotated_df[['Total']].copy()

# Group by part after first underscore in index
combined_df['Group'] = combined_df.index.str.split('_', n=1).str[1]
grouped_df = combined_df.groupby('Group').sum().reset_index()
combined_df = pd.concat([combined_df.drop(columns=['Group']),
grouped_df.set_index('Group')])

# Remove rows with specific prefixes
prefixes_to_remove = ('POL', 'MOL', 'FOL', 'F', 'M')
filtered_df =
combined_df[~combined_df.index.str.startswith(prefixes_to_remove)].cop
y()

# Keep only 'Tot_Tot' rows or single entries for each first word in
index
```

```python
filtered_df['First_Word'] = filtered_df.index.str.split('_').str[0]
indices_to_keep = []
for word in filtered_df['First_Word'].unique():
    group = filtered_df[filtered_df['First_Word'] == word]
    if len(group) > 1 and (group.index.str.endswith('Tot_Tot')).any():

indices_to_keep.append(group[group.index.str.endswith('Tot_Tot')].inde
x[0])
    else:
        indices_to_keep.append(group.index[0])

filtered_df = filtered_df.loc[indices_to_keep]
filtered_df =
filtered_df[~filtered_df.index.str.startswith(prefixes_to_remove)]
filtered_df = filtered_df.drop(columns=['First_Word'])

# Remove specific rows
rows_to_remove = ['Tot_Tot', 'P_Tot_Tot', 'Japan_Tot', 'LSatH_NS_Tot']
filtered_df = filtered_df.drop(index=rows_to_remove, errors='ignore')

# Simplify index to first word
filtered_df.index = filtered_df.index.str.split('_').str[0]

# Rename specific rows
row_rename_mapping = {
    'CL': 'Chinese Language',
    'AIndLng': 'Australian Indigenous Languages',
    'IAL': 'Indo-Aryan Languages',
    'Oth': 'Others',
    'SAL': 'Southeast Asian Austronesian Languages',
    'PSEO': 'Person that Speaks English Only'
}

filtered_df = filtered_df.rename(index=row_rename_mapping)

print(filtered_df)
```

```
                                             Total
Person that Speaks English Only         17020417.0
Australian Indigenous Languages           129528.0
Afrikaans                                  87482.0
Arabic                                    643449.0
Chinese Language                         1855888.0
Croatian                                  113772.0
Dutch                                      67671.0
German                                    158709.0
Greek                                     475176.0
Indo-Aryan Languages                     1238472.0
Italian                                   543195.0
Japanese                                   88905.0
```

```
Korean                                      217995.0
Others                                     1482220.0
Persian                                     116622.0
Polish                                       96159.0
Russian                                     100636.0
Southeast Asian Austronesian Languages      558917.0
Samoan                                       89737.0
Serbian                                     107601.0
Spanish                                     281630.0
Tamil                                       146320.0
Thai                                        110885.0
Turkish                                     116710.0
Vietnamese                                  554801.0
```

7.2.2 Language Data Cleaning (New Zealand 2013 and 2018)

Issues:

1. **Irrelevant Columns**: The dataset had unnecessary columns unrelated to language data.
2. **Missing Values**: Some rows lacked data in the `Language` and `Count` columns.
3. **Decimal Points**: Population counts had unnecessary decimal points.

Solutions:

1. **Column Filtering**: Retained only `Year`, `Language`, and `Count`.
2. **Dropped Missing Data**: Removed rows with missing values in key columns.
3. **Converted to Integers**: Removed decimal points by converting counts to integers.

```python
nz_language_data = pd.read_csv('nz_language_2013_2018.csv')

# Clean the language data by selecting relevant columns
nz_lang_clean = nz_language_data[['YEAR_CEN18_ECI_006', 'Languages
spoken', 'OBS_VALUE']].copy()

# Renaming columns
nz_lang_clean.columns = ['Year', 'Language', 'Count']

# Drop rows where Language or Count is NaN (missing values)
nz_lang_clean.dropna(subset=['Language', 'Count'], inplace=True)

# Filter to include only the years 2013 and 2018
nz_lang_clean = nz_lang_clean[nz_lang_clean['Year'].isin([2013,
2018])]

# Convert the Count column to integers to remove decimal points
nz_lang_clean['Count'] = nz_lang_clean['Count'].astype(int)

# View the cleaned data
print(nz_lang_clean)
```

```
    Year                     Language    Count
0   2013                      English  3819972
1   2018                      English  4482132
2   2013                        Maori   148395
3   2018                        Maori   185955
4   2013                       Samoan    86403
5   2018                       Samoan   101937
6   2013             Northern Chinese    52263
7   2018             Northern Chinese    95253
8   2013                        Hindi    66309
9   2018                        Hindi    69471
10  2013                       French    49125
11  2018                       French    55116
12  2013                          Yue    44625
13  2018                          Yue    52767
14  2013  Sinitic not further defined    42750
15  2018  Sinitic not further defined    51501
16  2013                      Tagalog    29016
17  2018                      Tagalog    43278
18  2013                       German    36642
19  2018                       German    41385
20  2013                      Spanish    26979
21  2018                      Spanish    38823
22  2013                    Afrikaans    27387
23  2018                    Afrikaans    36966
24  2013                       Tongan    31839
25  2018                       Tongan    35820
26  2013                      Panjabi    19749
27  2018                      Panjabi    34227
28  2013    New Zealand Sign Language    20235
29  2018    New Zealand Sign Language    22986
30  2013                        Other   265563
31  2018                        Other   349683
```

7.2.3 Australian 2016 Religion Data Cleaning

Issues:

1. **Multiple Columns**: The dataset included many columns, but we only needed total population columns ending with `'P'`.
2. **Unwanted Summary Column**: The `'Tot_P'` column was an overall population summary that needed to be excluded.
3. **Data Orientation**: Population data was in columns, but needed to be transposed for easier analysis.
4. **Sorting**: Religions were not sorted by population, making it harder to identify the largest groups.

Solutions:

1. **Filter Columns**: Used `filter(regex='P$')` to select only columns ending with `'P'`.

2. **Exclude Summary**: Dropped the `'Tot_P'` column to avoid duplication.
3. **Transpose Data**: Transposed the DataFrame with `.transpose()` to make religions the rows.
4. **Sort by Population**: Sorted the data in descending order using `.sort_values()`.

```
religionau2016 = pd.read_csv('aur_religion_2016.csv')

# Filter columns that end with 'P' and drop 'Tot_P'
filtered_columns = religionau2016.filter(regex='P$')

# Function to keep only the 'Tot_P' columns when there are multiple
with the same prefix
def keep_tot_p_only(df):
    unique_columns = {}
    for col in df.columns:
        prefix = col.split('_')[0]
        if prefix not in unique_columns or col.endswith('Tot_P'):
            unique_columns[prefix] = col
    return df[unique_columns.values()]

# Apply the function to filter out non-'Tot_P' columns
filtered_columns = keep_tot_p_only(filtered_columns)

# Rotate (transpose) the filtered DataFrame
rotated_filtered_columns = filtered_columns.transpose()

# Sort the transposed DataFrame by the values in descending order
sorted_rotated = rotated_filtered_columns.sort_values(by=0,
ascending=False)

# Remove specific rows by index
rows_to_remove = ['Christinty_Jehvahs_Witnses_P',
'Christnty_Sevnth_dy_Advntst_P', 'Othr_Rel_Aust_Abor_Trad_Rel_P']
sorted_rotated = sorted_rotated.drop(index=rows_to_remove)

# Rename specific rows
row_rename_mapping = {
    'Christianity_Tot_P': 'Christianity',
    'SB_OSB_NRA_Tot_P': 'Secular Beliefs and Other Spritual Beliefs
and No Religious Affiliation',
    'Religious_affiliation_ns_P': 'Religious Affiliation Not Stated',
    'Islam_P': 'Islam',
    'Buddhism_P': 'Buddhism',
    'Hinduism_P': 'Hinduism',
    'Other_Religions_Tot_P': 'Other Religions',
    'Judaism_P': 'Judaism'
}

sorted_rotated = sorted_rotated.rename(index=row_rename_mapping)

final_religion = sorted_rotated[1:]
```

```python
# Display the renamed DataFrame
print(final_religion)
```

```
                                                        0
Christianity                                     11148814
Secular Beliefs and Other Spritual Beliefs and ...  9886957
Religious Affiliation Not Stated                  1848426
Islam                                              813392
Hinduism                                           684002
Buddhism                                           615823
Other Religions                                    325421
Judaism                                             99956
```

New Zealand 2013 & 2018 Religion Data Cleaning

Issues:

1. **Multiple Irrelevant Categories:** The dataset included summary categories such as `'Total'`, `'Total stated'`, `'Not elsewhere included'`, and `'Object to answering'`, which needed to be excluded for cleaner analysis.
2. **Non-numeric Values**: The `'Census_usually_resident_population_count'` column contained non-numeric values (e.g., `'..'`), which interfered with numerical analysis.
3. **Duplicate Entries for Some Religions**: Some religions had multiple rows for the same year with different population counts, which required aggregation.
4. **Unnecessary Year Range**: The dataset included multiple years, but we only needed data from 2013 and 2018.

Solutions:

1. **Exclude Irrelevant Categories**: Filtered out summary categories (`'Total'`, `'Total stated'`, `'Not elsewhere included'`, and `'Object to answering'`) to focus solely on individual religious groups.
2. **Handle Non-numeric Values:** Converted the `'Count'` column to numeric using `pd.to_numeric()` with `errors='coerce'` to replace non-numeric values with `NaN`, then dropped these rows.
3. **Aggregate Data**: Grouped the dataset by `'Year'` and `'Religion'` and summed the population counts to consolidate multiple entries for the same religion in the same year.
4. **Filter Relevant Years:** Selected only the 2013 and 2018 data using `.isin([2013, 2018])` to focus on those years.

```python
# Select relevant columns (Year, Religion, Count)
nz_rel_clean = nz_religion[['Year',
'Religious_affiliation_description',
'Census_usually_resident_population_count']].copy()

# Rename columns for clarity
nz_rel_clean.columns = ['Year', 'Religion', 'Count']
```

```
# Drop rows where 'Religion' or 'Count' is missing (i.e., NaN values)
nz_rel_clean.dropna(subset=['Religion', 'Count'], inplace=True)

# Convert 'Count' column to numeric, replacing non-numeric values
(like '..') with NaN
nz_rel_clean['Count'] = pd.to_numeric(nz_rel_clean['Count'],
errors='coerce')

# Drop rows where 'Count' is NaN after the conversion
nz_rel_clean.dropna(subset=['Count'], inplace=True)

# Convert the 'Count' column to integer
nz_rel_clean['Count'] = nz_rel_clean['Count'].astype(int)

# Filter data for only the years 2013 and 2018
nz_rel_clean = nz_rel_clean[nz_rel_clean['Year'].isin([2013, 2018])]

# Sort the data by 'Year' and 'Religion' for easier viewing
nz_rel_clean = nz_rel_clean.sort_values(by=['Year',
'Religion']).reset_index(drop=True)

# Aggregate the data by 'Year' and 'Religion' by summing the 'Count'
values
nz_rel_clean = nz_rel_clean.groupby(['Year', 'Religion'],
as_index=False)['Count'].sum()

# Sort the aggregated data again by 'Year' and 'Religion'
nz_rel_clean = nz_rel_clean.sort_values(by=['Year',
'Religion']).reset_index(drop=True)

# List of categories to exclude (e.g., 'Total', 'Not elsewhere
included')
exclude_categories = [
    'Total', 'Total stated', 'Not elsewhere included', 'Object to
answering'
]

# Remove rows where the 'Religion' column contains any of the excluded
categories
nz_rel_clean =
nz_rel_clean[~nz_rel_clean['Religion'].isin(exclude_categories)]

# Display the final cleaned
print(nz_rel_clean)
```

```
    Year                                   Religion     Count
0   2013                                   Buddhism    716718
1   2013                                  Christian  23074854
2   2013                                   Hinduism   1089033
3   2013                                      Islam    563955
```

```
4   2013                                        Judaism     85749
5   2013  Māori religions, beliefs and philosophies    715908
6   2013                                    No religion  20547561
9   2013  Other religions, beliefs and philosophies    389349
10  2013         Spiritualism and New Age religions    232236
13  2018                                       Buddhism    645702
14  2018                                      Christian  21486354
15  2018                                       Hinduism   1493634
16  2018                                          Islam    743763
17  2018                                        Judaism     65985
18  2018  Māori religions, beliefs and philosophies    799578
19  2018                                    No religion  28657842
22  2018  Other religions, beliefs and philosophies   1130400
23  2018         Spiritualism and New Age religions    248898
```

8. Exploratory Data Analysis (EDA)

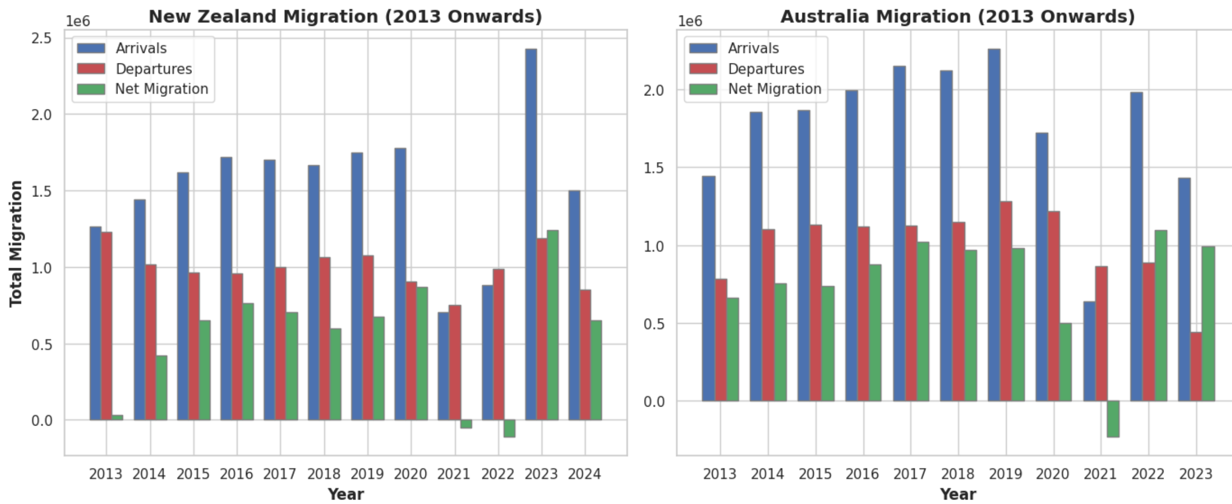
8.1 **Migration Data Visualization**

Datasets:

- `migrationau.csv` (Australia Migration Data)
- `migrationnz.csv` (New Zealand Migration Data)

```
# New Zealand: Extract the year and group by 'Year'
nz_migration_cleaned['Year'] =
pd.to_datetime(nz_migration_cleaned['Date'], format='%b-%Y').dt.year
nz_migration_filtered =
nz_migration_cleaned[nz_migration_cleaned['Year'] >= 2013]
yearly_data_nz = nz_migration_filtered.groupby('Year')[['Migrant
Arrivals', 'Migrant Departures', 'Net Migration']].sum()

# Australia: Extract the year and group by 'Year'
aus_migration_cleaned['Year'] =
pd.to_datetime(aus_migration_cleaned['Date'], format='%b-%Y').dt.year
aus_migration_filtered =
aus_migration_cleaned[aus_migration_cleaned['Year'] >= 2013]
yearly_data_aus = aus_migration_filtered.groupby('Year')[['Migrant
Arrivals', 'Migrant Departures', 'Net Migration']].sum()

# Create subplots
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 6))

# Define bar width and positions on the x-axis for both graphs
bar_width = 0.25

# Plot for New Zealand
years_nz = yearly_data_nz.index
r1_nz = range(len(years_nz))
r2_nz = [x + bar_width for x in r1_nz]
```

```
r3_nz = [x + bar_width for x in r2_nz]

axes[0].bar(r1_nz, yearly_data_nz['Migrant Arrivals'], color='b',
width=bar_width, edgecolor='grey', label='Arrivals')
axes[0].bar(r2_nz, yearly_data_nz['Migrant Departures'], color='r',
width=bar_width, edgecolor='grey', label='Departures')
axes[0].bar(r3_nz, yearly_data_nz['Net Migration'], color='g',
width=bar_width, edgecolor='grey', label='Net Migration')
axes[0].set_title('New Zealand Migration (2013 Onwards)',
fontweight='bold', fontsize=14)
axes[0].set_xlabel('Year', fontweight='bold')
axes[0].set_ylabel('Total Migration', fontweight='bold')
axes[0].set_xticks([r + bar_width for r in range(len(years_nz))])
axes[0].set_xticklabels(years_nz)
axes[0].legend()

# Plot for Australia
years_aus = yearly_data_aus.index
r1_aus = range(len(years_aus))
r2_aus = [x + bar_width for x in r1_aus]
r3_aus = [x + bar_width for x in r2_aus]

axes[1].bar(r1_aus, yearly_data_aus['Migrant Arrivals'], color='b',
width=bar_width, edgecolor='grey', label='Arrivals')
axes[1].bar(r2_aus, yearly_data_aus['Migrant Departures'], color='r',
width=bar_width, edgecolor='grey', label='Departures')
axes[1].bar(r3_aus, yearly_data_aus['Net Migration'], color='g',
width=bar_width, edgecolor='grey', label='Net Migration')
axes[1].set_title('Australia Migration (2013 Onwards)',
fontweight='bold', fontsize=14)
axes[1].set_xlabel('Year', fontweight='bold')
axes[1].set_xticks([r + bar_width for r in range(len(years_aus))])
axes[1].set_xticklabels(years_aus)
axes[1].legend()

# Adjust layout and show the plot
plt.tight_layout()
plt.show()
```



In New Zealand, net migration rose from 2013 to 2016, fluctuated in 2017, and hit a record high in 2019. Despite the pandemic in 2020, migrant arrivals remained high, likely due to short-term visitors classified as migrants when borders closed. In 2021 and 2022, net migration turned negative with significantly fewer arrivals. By 2023, migrant arrivals in New Zealand reached a new record high.

In Australia, migration arrivals remained consistently high at around 500,000, with positive net migration each year. Net migration turned negative in 2021, but following the pandemic, migrant arrivals reached an all-time high. Overall, there is large flow of migrants into the country, which contributes to cultural diversity.

Australia consistently has higher migrant arrivals than New Zealand, suggesting that Australia might have a larger immigrant population. In both countries, there was the net migration was positive except for special circumstances (COVID-19). Hence, we conclude that the migrant population is growing and makes up a significant proportion of the countries' population.

Comparison:

1. Australia's migration numbers significantly larger than New Zealand's.
2. Both experienced COVID-19 impact, but Australia shows clearer recovery pattern.
3. Australia facing record-high post-COVID migration, while New Zealand projects more moderate levels.

Key Implications:

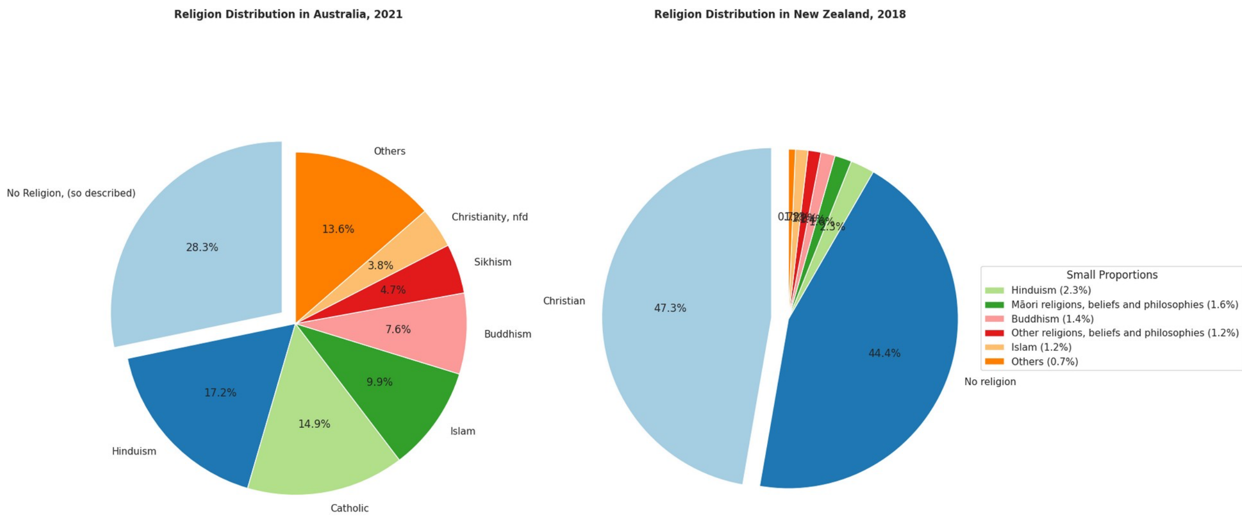
1. Continued contribution to multicultural societies in both countries.
2. Australia may face near-term challenges in accommodating rapid population growth.

8.2 Religion Data Visualisation

```python
# Assuming top_8_aus_religions and top_8_nz_religions are already defined

# Data for Australian religion pie chart
religions_aus = top_8_aus_religions['Religion']
```

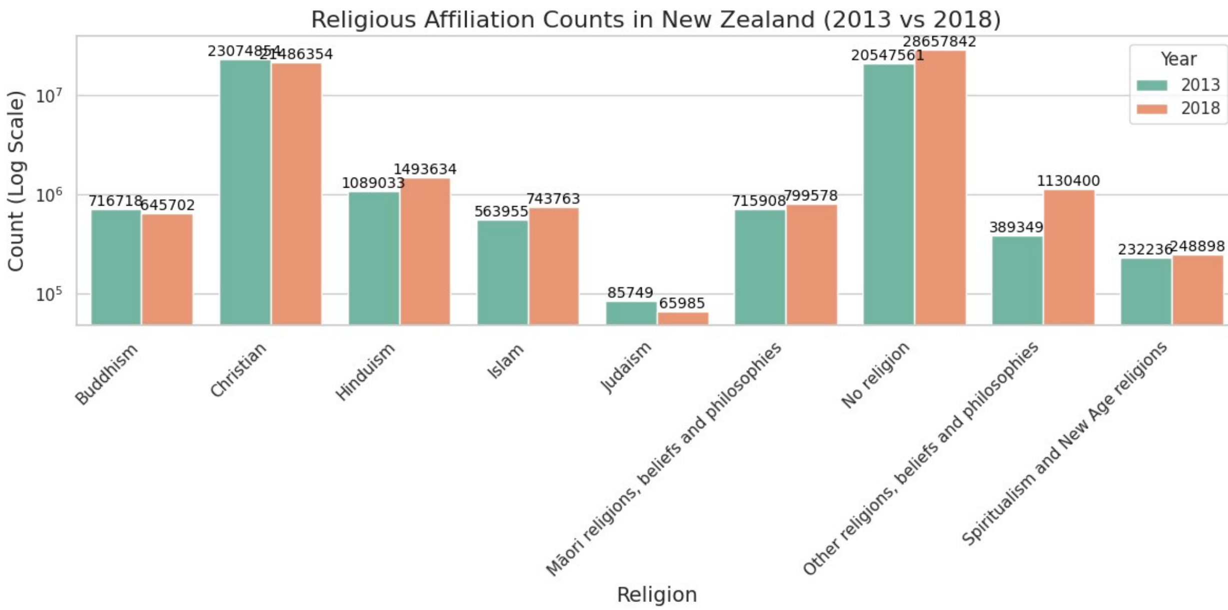
```
counts_aus = top_8_aus_religions['Count']

# Data for New Zealand religion pie chart
religions_nz = top_8_nz_religions['Religion']
counts_nz = top_8_nz_religions['Count']

# Create subplots: one row and two columns
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 10))  # Increased
figure size

# Pie chart for Australia
explode_aus = [0.1 if religion == 'No Religion, (so described)' else 0
for religion in religions_aus]
ax1.pie(counts_aus, labels=religions_aus, autopct='%1.1f%%',
startangle=90, explode=explode_aus, colors=plt.cm.Paired.colors)
ax1.set_title('Religion Distribution in Australia, 2021',
fontweight='bold')

# Pie chart for New Zealand
explode_nz = [0.1 if religion == 'Christian' else 0 for religion in
religions_nz]
wedges, texts, autotexts = ax2.pie(counts_nz, autopct='%1.1f%%',
startangle=90, explode=explode_nz, colors=plt.cm.Paired.colors)

# Only label slices with more than 3% representation directly on the
pie
threshold = 3
for i, (pct, religion) in enumerate(zip(autotexts, religions_nz)):
    if float(pct.get_text()[:-1]) > threshold:
        texts[i].set_text(religion)
    else:
        texts[i].set_text('')

# Add a legend for the smaller slices
legend_labels = [f'{religion} ({pct.get_text()})' for religion, pct in
zip(religions_nz, autotexts) if float(pct.get_text()[:-1]) <=
threshold]
ax2.legend(wedges[-len(legend_labels):], legend_labels, title="Small
Proportions", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))

ax2.set_title('Religion Distribution in New Zealand, 2018',
fontweight='bold')

# Ensure equal aspect ratio for both pie charts
ax1.axis('equal')
ax2.axis('equal')

# Adjust layout
plt.tight_layout()
```

```python
# Show the plot
plt.show()
```



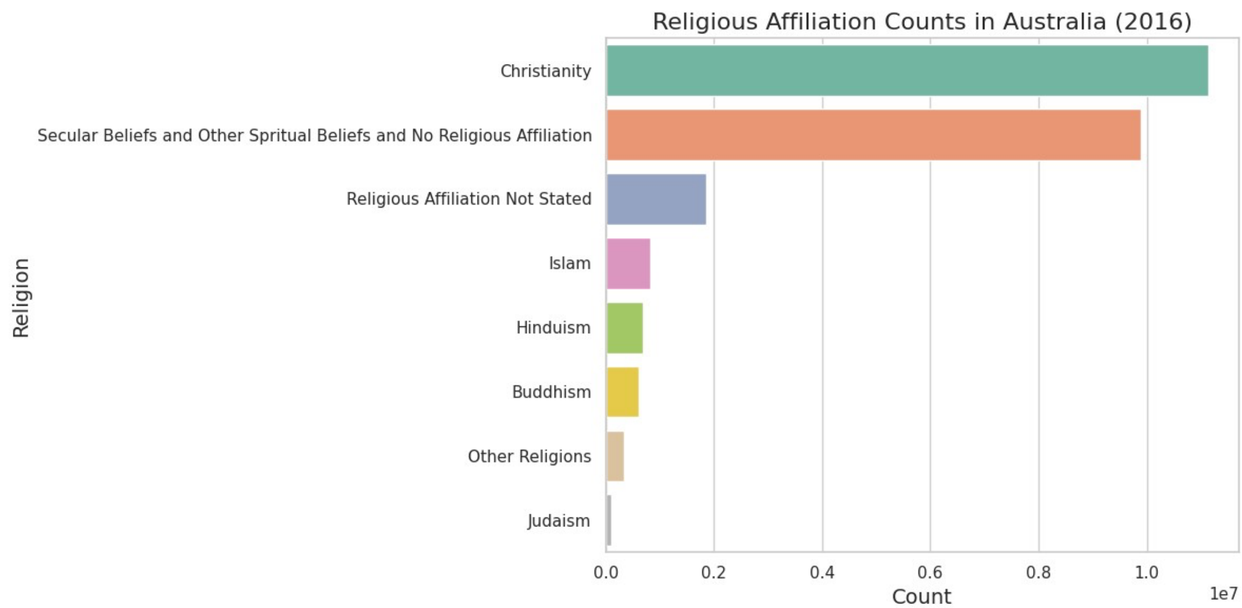

```python
import warnings

# Suppress the specific FutureWarning
warnings.filterwarnings("ignore", category=FutureWarning,
module="seaborn")

# Set the style for the plot
sns.set(style='whitegrid')

# Create the barplot
plt.figure(figsize=(12, 6))  # Set the figure size

# Create a barplot with a logarithmic scale on the y-axis
sns.barplot(x='Religion', y='Count', hue='Year', data=nz_rel_clean,
palette='Set2')

# Set y-axis to log scale
plt.yscale('log')

# Add plot labels and title
plt.title('Religious Affiliation Counts in New Zealand (2013 vs
2018)', fontsize=16)
plt.xlabel('Religion', fontsize=14)
plt.ylabel('Count (Log Scale)', fontsize=14)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')
```

```
# Add data labels on top of the bars
for p in plt.gca().patches:
    plt.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width()
/ 2., p.get_height()),
                 ha='center', va='bottom', fontsize=10, color='black',
rotation=0)

# Show the plot
plt.tight_layout()
plt.show()
```




```
#  Reset the index to make Religion names a column
final_religion = final_religion.reset_index()

#  Rename the columns for clarity
final_religion.columns = ['Religion', 'Count']

#  Set the style for the plot
sns.set(style='whitegrid')

# : Create the bar plot
plt.figure(figsize=(12, 6))  # Set the figure size

# Create a horizontal bar plot
sns.barplot(x='Count', y='Religion', data=final_religion,
palette='Set2')

# Add plot labels and title
plt.title('Religious Affiliation Counts in Australia (2016)',
fontsize=16)
plt.xlabel('Count', fontsize=14)
```

```
plt.ylabel('Religion', fontsize=14)

# Show the plot
plt.tight_layout()
plt.show()
```



In Australia, Christianity was the largest religious group in both 2016 and 2021, but its number of adherents declined. In contrast, secular beliefs grew significantly and formed the second largest group in 2021. Additionally, the populations of Islam, Buddhism, and Hinduism increased, highlighting the country's growing diversity.

In New Zealand, Christianity was the largest religious affiliation in 2013 but was surpassed by secular beliefs in 2018. There was also significant growth in Hinduism, Sikhism, and Islam, indicating a wave of migration from India, alongside a rise in those identifying as Jedi.

Both countries are experiencing secularization, with Christianity remaining dominant. The increased presence of Hinduism, Buddhism, and Islam reflects migration from Asia and the Middle East.

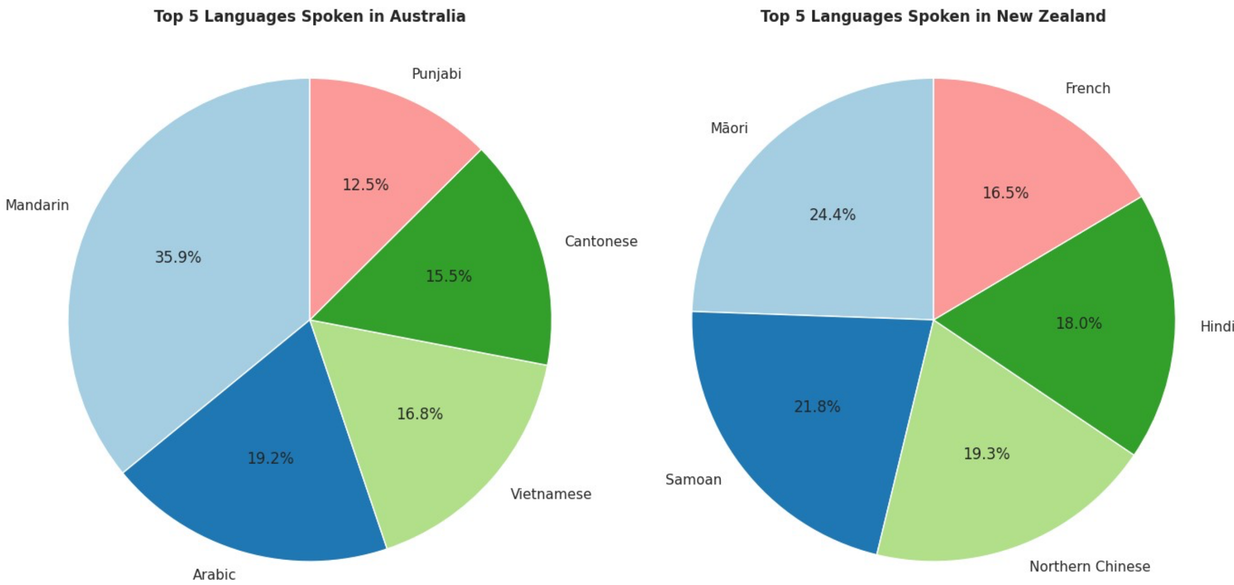
8.3 Languages Spoken Data Visualisation

In Australia, Chinese languages (CL) encompass all dialects, including Mandarin, Cantonese, Hakka, Wu, and Min Nan.

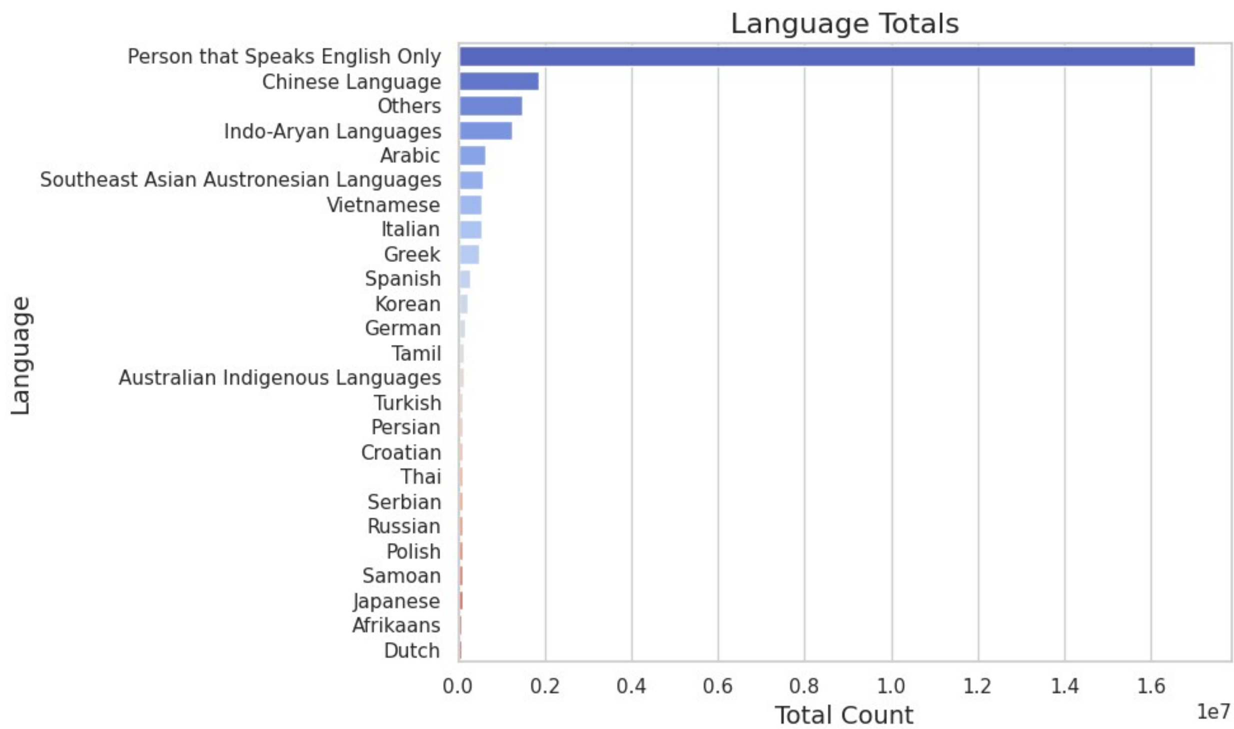
The category 'Other' includes languages not specifically identified, inadequately described, or non-verbal.

Indo-Aryan languages (IAL) include Bengali, Hindi, Punjabi, Sinhalese, Urdu, Gujarati, Konkani, Marathi, Nepali, Sindhi, Assamese, Dhivehi, Kashmiri, Oriya, Fijian Hindustani, and others. Southeast Asian Austronesian languages (SAL) include Filipino, Indonesian, Tagalog, Bikol, Bisaya, Cebuano, Ilokano, Ilonggo, Pampangan, Malay, Tetum, Timorese, Acehnese, Balinese, Iban, Javanese, and more.

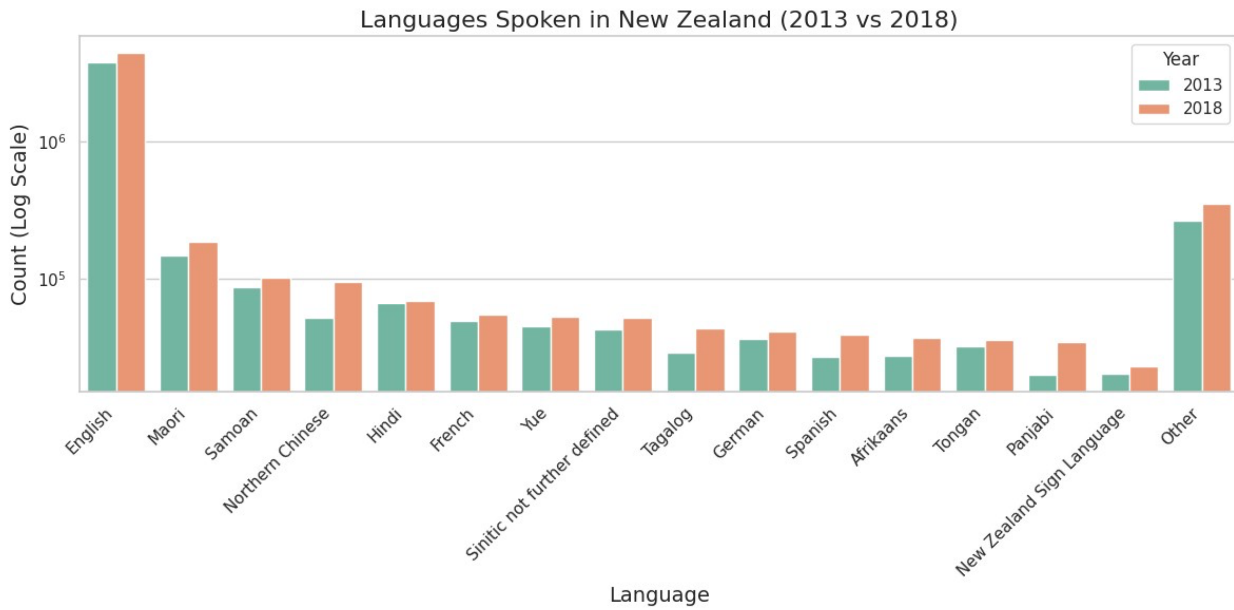
```python
# Data for Australian language pie chart
languages_aus = aus_language_cleaned['Language']
counts_aus = aus_language_cleaned['Count']

# Data for New Zealand language pie chart
languages_nz = nz_top5_languages['Language']
counts_nz = nz_top5_languages['Count']

# Create subplots: one row and two columns
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 7))

# Pie chart for Australia
ax1.pie(counts_aus, labels=languages_aus, autopct='%1.1f%%',
startangle=90, colors=plt.cm.Paired.colors)
ax1.set_title('Top 5 Languages Spoken in Australia',
fontweight='bold')

# Pie chart for New Zealand
ax2.pie(counts_nz, labels=languages_nz, autopct='%1.1f%%',
startangle=90, colors=plt.cm.Paired.colors)
ax2.set_title('Top 5 Languages Spoken in New Zealand',
fontweight='bold')

# Ensure equal aspect ratio for both pie charts
ax1.axis('equal')
ax2.axis('equal')

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```




```python
# Australian variation file 2016

# Set the style for the plot
sns.set(style="whitegrid")

# Reset the index to make the language names accessible as a column
filtered_df = filtered_df.reset_index()

# Rename columns for clarity
filtered_df.columns = ['Language', 'Total']

# Sort the DataFrame by 'Total' for a better visualization
filtered_df = filtered_df.sort_values(by='Total', ascending=False)

# Create a bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x='Total', y='Language', data=filtered_df,
palette='coolwarm')

# Add titles and labels
plt.title('Language Totals', fontsize=16)
plt.xlabel('Total Count', fontsize=14)
plt.ylabel('Language', fontsize=14)

# Display the plot
plt.tight_layout()
plt.show()
```




```
# New Zealand variation file from 2013 and 2018

# Set the style for the plot
sns.set(style='whitegrid')

# Create a bar plot
plt.figure(figsize=(12, 6))  # Set the figure size

# Create a bar plot with separate bars for each year
sns.barplot(x='Language', y='Count', hue='Year', data=nz_lang_clean,
palette='Set2')

# Set y-axis to log scale
plt.yscale('log')

# Add plot labels and title
plt.title('Languages Spoken in New Zealand (2013 vs 2018)',
fontsize=16)
plt.xlabel('Language', fontsize=14)
plt.ylabel('Count (Log Scale)', fontsize=14)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Show the plot
plt.tight_layout()
plt.show()
```



In Australia, a large proportion of the population speaks only English, with Mandarin being the most common language spoken at home other than English. Between 2016 and 2021, there was a notable rise in the proportion of people speaking Punjabi, Mandarin, Nepali, and Arabic (Chinese and Indo-Aryan languages), highlighting the growing linguistic diversity and the influence of migration from Asia and the Middle East.

In New Zealand, English is the dominant language, followed by Māori and Samoan. The increase in Māori and Samoan speakers between 2013 and 2018 suggests that efforts to celebrate and preserve the languages of the indigenous and Pacific communities have been successful. Other widely spoken languages include Hindi, Chinese dialects (such as Yue, spoken in Hong Kong), and Tagalog, reflecting significant Asian immigration. European languages like French, German, and Spanish are also present, alongside Tongan, an Austronesian language, indicating the influence of Pacific Island cultures.

In both countries, the presence of Chinese and Indo-Aryan languages further illustrates the strong Chinese and Indian communities contributing to the multicultural landscapes of Australia and New Zealand.

These linguistic trends highlight the increasing cultural diversity, emphasizing the importance of creating an environment that accommodates speakers of various languages. The growth of Polynesian languages, in particular, underscores the potential for inclusive policies that support and celebrate multilingual communities.

9. Discussion

This study emphasizes the ongoing migration to Australia and New Zealand, which contributes to their cultural diversity, examined here through the lenses of languages spoken and religious affiliations. Although the data collection methods for languages spoken differ between the two countries, they remain comparable as they provide a representation of the primary languages spoken by the population. Despite variations in census years—Australian data from 2021 and 2016, and New Zealand data from 2018 and 2013—both sets reflect changes and growth in the population over five-year periods. This allows us to highlight shifts in cultural dynamics and compare them between the two nations. Our key findings include:

1) There is a large proportion of Chinese and Indian migrants into both Australia and New Zealand. These findings are supported by the census data on religion and language.

2) The successful revival of Polynesian languages in New Zealand serves as a model for other countries to celebrate and preserve their native populations.

Suggestions for moving forward:

1) Creation of more inclusive environments, such as strategic placement of religious infrastructure, translation services in essential sectors, language inclusivity in media

2) Integration policies to ensure social cohesion, such as programs to promote cross-cultural understanding

3) Language and cultural programs to support new migrants

4) Education system adaptations to reflect diverse student backgrounds and to support language maintenance