STEP 1: Import libraries and read the dataset

In [1]:

```
# import libraries
import pandas as pd
import seaborn as sns
import numpy as np

sns.set_theme()
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
# Read the data
census_data = pd.read_csv('census_data.csv')
```

STEP 2: Study the dataset

In [3]:

```
# print out the first five rows to have an overview of the dataset
census_data.head()
```

Out[3]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation	I
0	1	George Avenue	Harry	James	60	Head	Single	Male	Unemployed	
1	2	George Avenue	Anne	Johnson	34	Head	Married	Female	Corporate treasurer	
2	2	George Avenue	Jack	Johnson	36	Husband	Married	Male	Product/process development scientist	
3	2	George Avenue	Guy	Johnson	12	Son	NaN	Male	Student	
4	3	George Avenue	Simon	Smith	79	Head	Single	Male	Retired Tour manager	

In [4]:

```
# To check the number of rows and columns in the dataset
census_data.shape
```

Out[4]:

```
(8646, 11)
```

In [5]:

```
#Displays the data type, and number of entries of the data
census_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8646 entries, 0 to 8645
Data columns (total 11 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   House Number                    8646 non-null   int64 
 1   Street                          8646 non-null   object
 2   First Name                      8646 non-null   object
 3   Surname                         8646 non-null   object
 4   Age                             8646 non-null   object
 5   Relationship to Head of House   8646 non-null   object
 6   Marital Status                  6419 non-null   object
 7   Gender                          8646 non-null   object
 8   Occupation                      8646 non-null   object
 9   Infirmity                       8646 non-null   object
 10  Religion                        6373 non-null   object
dtypes: int64(1), object(10)
memory usage: 743.1+ KB
```

In [6]:

```
# Check the total number of missing values
census_data.isna().sum()
```

Out[6]:

```
House Number                       0
Street                             0
First Name                         0
Surname                            0
Age                                0
Relationship to Head of House      0
Marital Status                  2227
Gender                             0
Occupation                         0
Infirmity                          0
Religion                        2273
dtype: int64
```

In [7]:

```
# Check for duplicate
dupli = census_data.duplicated()
census_data[dupli]
```

Out[7]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupatior
7309	1	Leedsbox Crescent	Ashleigh	Osborne	15	Daughter	NaN	Female	Studen

In [8]:

```
# Drop the duplicate row
census_data = census_data.drop(7309)
```

STEP 3: Data Cleaning

Religion attribute

In [9]:

```
# Check the unique entries in Religion
print(census_data['Religion'].unique())
```

```
['None' nan 'Jewish' 'Catholic' 'Christian' 'Methodist' 'Muslim' 'Sikh'
 'Orthodoxy' 'Baptist' 'Undecided' 'Buddist' ' ' 'Nope']
```

In [10]:

```
# Replace 'Nope' with 'None' for consistency
census_data['Religion'].replace('Nope', 'None', regex = True, inplace = True)
```

In [11]:

```
# Check for blank entries in Religion
census_data[census_data['Religion'] == ' ']
```

Out[11]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation
7069	4	Parsons Stream	Neil	Hall	34	Husband	Married	Male	Mining engineer
7809	1	Cox Drive	Valerie	Arnold	57	Head	Single	Female	Unemployed
8385	57	George Lane	Debra	Davies	31	Head	Married	Female	Barista
8433	67	George Lane	Ashleigh	Martin	38	Lodger	Single	Female	Minerals surveyor

In [12]:

```
# Check if they have family members that also filled the form
census_data[7807:7812]
```

Out[12]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupa
7808	4	Kelly Mountain	Denise	Thompson	10	Daughter	NaN	Female	Stu
7809	1	Cox Drive	Valerie	Arnold	57	Head	Single	Female	Unemplc
7810	1	Cox Drive	Katie	Arnold	23	Daughter	Single	Female	He promc speci
7811	2	Cox Drive	Kyle	Perkins	72	Head	Widowed	Male	Ret Film/v e
7812	3	Cox Drive	Elizabeth	Steele	20	Head	Single	Female	Unive Stu

In [13]:

```
# I changed row 7809 to Christian, beacause her daughter is a Christian(entry 7810),
# it might be an ommision when she was filling the form.

census_data.at[7809,'Religion'] = 'Christian'
```

In [14]:

```
# I changed the remaining 3 entries to None, because religion is a sensitive attribute,
# I think it is not appropriate to assign any religion to someone.

census_data['Religion'].replace(' ', 'None', regex = True, inplace = True)
```

In [15]:

```
# Replace other missing values(nan) in Religion to None
census_data['Religion'] = census_data['Religion'].fillna('None')
```

Data Cleaning in Age attribute

In [16]:

```
# Check the unique entries in Age
print(census_data['Age'].unique())
```

```
['60' '34' '36' '12' '79' '35' '61' '24' '3' '75' '52' '14' '11' '42' '2
5'
 '28' '40' '57' '55' '22' '18' '43' '51' '0' '21' '45' '17' '16' '13' '9'
 '65' '32' '31' '8' '56' '39' '7' '41' '27' '78' '30' '29' '15' '54' '19'
 '84' '38' '33' '6' '1' '48' '10' '5' '49' '46' '26' '50' '53' '63' '4'
 '44' '47' '2' '23' '64' '37' '58' '66' '67' '71' '72' '20' '62' '68' '7
3'
 '74' '69' '81' '70' '59' '89' '105' '87' '80' '77' '76' ' ' '82' '88'
 '49.16040882016717' '54.16040882016717' '3.0' '85' '99' '101' '83'
 '69.13036593215614' '67.13036593215614' '103' '90' '93' '86' '96'
 '85.66111048772531' '87.66111048772531' '34.0' '30.0' '26.0' '91' '102'
 '83.52432893335205' '26.999999999999993' '23.999999999999993'
 '21.999999999999993' '16.999999999999993' '92' '97' '69.13473801820774'
 '15.000000000000007' '13.000000000000007' '10.000000000000007' '98'
 '50.53760781824045' '53.53760781824045' '0.0']
```

In [17]:

```
# Check for blank entries in Age
census_data[census_data['Age'] == ' ']
```

Out[17]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation
460	18	Smith Gateway	Dominic	Griffiths		Son	NaN	Male	Student

In [18]:

```
# I drop the row since it is just a row, it has little or no effect on the data
census_data = census_data.drop(460)
```

In [19]:

```
# Convert to integer
census_data['Age'] = census_data['Age'].astype(float).round(0).astype(int)
```

In [20]:

```
# Confirm that the missing values has been replaced
census_data['Age'].isna().sum()
```

Out[20]:

```
0
```

Data Cleaning for Marital Status attribute

In [21]:

```
# Check the unique entries in Marital Status
print(census_data['Marital Status'].unique())
```

```
['Single' 'Married' nan 'Divorced' 'Widowed' ' ']
```

In [22]:

```
# Check for blank entries
census_data[census_data['Marital Status'] == ' ']
```

Out[22]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation	In
3205	43	Morgan Fords	Diana	Robinson	39	Wife		Female	Hospital pharmacist	

In [23]:

```
# Check the range for any family member
census_data[3203:3209]
```

Out[23]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupatic
3204	43	Morgan Fords	Peter	Robinson	42	Head	Married	Male	Designe interior/spati
3205	43	Morgan Fords	Diana	Robinson	39	Wife		Female	Hospit pharmaci
3206	43	Morgan Fords	Wayne	Robinson	12	Son	NaN	Male	Stude
3207	43	Morgan Fords	Dale	Robinson	5	Son	NaN	Male	Stude
3208	43	Morgan Fords	Charles	Robinson	3	Son	NaN	Male	Chi
3209	44	Morgan Fords	Beverley	Williams	39	Head	Divorced	Female	Copywrite advertisi

In [24]:

```
# Replace the blank cell for line_num 3205 with 'Married' since the husband status is ma
census_data.at[3205,'Marital Status'] = 'Married'
```

In [25]:

```
# Replace the blank cell for line_num 3206:3208 with 'Single' since they are children.
census_data.at[3206,'Marital Status'] = 'Single'
census_data.at[3207,'Marital Status'] = 'Single'
census_data.at[3208,'Marital Status'] = 'Single'
```

In [26]:

```
# Substitute the marital status of individuals lesser than 18 years to 'Single'
children_age = census_data['Age'] < 18
census_data.loc[children_age,"Marital Status"] = census_data[children_age]["Marital Stat
```

In [27]:

```
# Fill the remaining missing values with 'None'
census_data['Marital Status'] = census_data['Marital Status'].fillna('None')
```

In [28]:

```
# Confirm that the missing values = 0
census_data['Marital Status'].isna().sum()
```

Out[28]:

```
0
```

Data Cleaning for Infirmity

In [29]:

```
print(census_data['Infirmity'].unique())
```

```
['None' 'Physical Disability' 'Mental Disability' ' ' 'Deaf' 'Blind'
 'Unknown Infection' 'Disabled']
```

In [30]:

```
# Check for blank entries
census_data[census_data['Infirmity']== ' ']
```

Out[30]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupa
556	1	Morgan View	Sean	Howe	15	Son	Single	Male	Stu
909	15	Newfound Station	Lynda	Murphy	24	Head	Single	Female	Public af consu
1120	24	Palmer Crescent	Garry	Burns	52	Husband	Married	Male	Act
4244	47	Madridgate Drive	Fiona	Lloyd	79	Wife	Married	Female	Re Contra
6047	12	Graham Road	Caroline	Bruce	46	Head	Divorced	Female	Chart manager accour
7727	9	Salmon Lane	Holly	Francis	40	Head	Single	Female	Program syst

In [31]:

```
# Replace the blank entries with 'None'
census_data['Infirmity']= census_data['Infirmity'].replace(' ', 'None')
```

Data Cleaning for Gender

In [32]:

```
# Check for blank entries
census_data[census_data['Gender']== ' ']
```

Out[32]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupatic
6013	9	Lime Street	Elizabeth	Dobson	4	Daughter	Single		Chi
7538	37	Leedsbox Crescent	Liam	Yates	66	Head	Married		Televisic productic assista

In [33]:

```
# Check range for any related information
census_data[7537:7541]
```

Out[33]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupa
7539	37	Leedsbox Crescent	Hayley	Yates	66	Wife	Married	Female	Soli Sco
7540	37	Leedsbox Crescent	Dylan	Yates	41	Son	Single	Male	Mate engi
7541	37	Leedsbox Crescent	Terry	Yates	38	Son	Single	Male	Desi fashion/clo
7542	37	Leedsbox Crescent	Annette	Yates	36	Daughter	Divorced	Female	

In [34]:

```
# Check range for any related information
census_data[6010:6015]
```

Out[34]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation
6011	9	Lime Street	Josephine	Dobson	32	Head	Married	Female	Corporate investment banker
6012	9	Lime Street	Francis	Dobson	31	Husband	Married	Male	IT sales professional
6013	9	Lime Street	Elizabeth	Dobson	4	Daughter	Single		Child
6014	10	Lime Street	Oliver	Willis	48	Head	Married	Male	Herpetologist
6015	10	Lime Street	Carole	Willis	42	Wife	Married	Female	Textile designer

In [35]:

```
# Replaced with Male, since he is the husband from his household
census_data.at[7538,'Gender'] = 'Male'

# Replace with Female, since her status is 'Daughter'
census_data.at[6013,'Gender'] = 'Female'
```

Data Cleaning for Surname

In [36]:

```
# Check for blank entries
census_data[census_data['Surname']== ' ']
```

Out[36]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation
2123	24	Morley Lodge	Simon		56	None	Single	Male	Information officer
3168	33	Morgan Fords	Stephanie		8	Daughter	Single	Female	Student

In [37]:

```
# Check range for any family related information
census_data[2120:2125]
```

Out[37]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupation
2121	24	Morley Lodge	Gary	Burton	29	None	Single	Male	Engineer maintenance (IT
2122	24	Morley Lodge	Caroline	Barber	38	None	Single	Female	Pharmacologis
2123	24	Morley Lodge	Simon		56	None	Single	Male	Information office
2124	25	Morley Lodge	Dylan	Griffiths	34	Head	Single	Male	Financia manage
2125	25	Morley Lodge	Eleanor	Griffiths	43	Cousin	Single	Female	Teacher secondary schoo

In [38]:

```
# Check range for any family related information
census_data[3165:3170]
```

Out[38]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupatic
3166	33	Morgan Fords	Lorraine	Griffin	31	Head	Married	Female	Unemploye
3167	33	Morgan Fords	Henry	Griffin	31	Husband	Married	Male	Acupuncturi
3168	33	Morgan Fords	Stephanie		8	Daughter	Single	Female	Stude
3169	33	Morgan Fords	Francis	Griffin	4	Son	Single	Male	Chi
3170	33	Morgan Fords	Kathryn	Dobson	2	Daughter	Single	Female	Chi

In [39]:

```
# Ignore row 2123 and leave it blank since he does not have any family member

# Change 3168 to Griffin, since  are family name is 'Griffin'
census_data.at[3168, 'Surname']='Griffin'
```

Data Cleaning for First Name

In [40]:

```
# Check for blank cells in First Name
census_data[census_data['First Name']== ' ']

#First Name is a unique value, I will ignore it and continue, as this will not affect ou
```

Out[40]:

	House Number	Street	First Name	Surname	Age	Relationship to Head of House	Marital Status	Gender	Occupati
618	19	Morgan View		Ali	8	Son	Single	Male	Stude
3266	4	Simmons Course		Wong	9	Son	Single	Male	Stude
3916	6	ExcaliburBells Road		Doyle	5	Son	Single	Male	Stude

Check for blanks for other attributes

In [41]:

```
len(census_data[census_data['House Number']== ' '])
```

Out[41]:

```
0
```

In [42]:

```
len(census_data[census_data['Street'] ==' '])
```

Out[42]:

```
0
```

In [43]:

```
len(census_data[census_data['Relationship to Head of House']== ' '])
```

Out[43]:

```
0
```

In [44]:

```
len(census_data[census_data['Occupation']== ' '])
```

Out[44]:

```
0
```

Check for lies in Age

In [45]:

```
# Check Maximum Age, to identify if there is any lie
census_data['Age'].max()
```

Out[45]:

```
105
```

In [46]:

```
# Check Minimum Age, to identify if there is any lie
census_data['Age'].min()
```

Out[46]:

```
0
```

In [47]:

```
# Filter the Married entries in Marital status to identify if anyone below 18years is ma
adult = census_data['Marital Status'].isin(['Married'])
child = adult & census_data['Age'].isin([census_data['Age']<18])
len(census_data[child])
```

Out[47]:

```
0
```