

```
In [1]:import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

DATA CLEANING

```
In [2]:# Reading the CENSUS Data
```

```
data=pd.read_csv("census13.csv")
```

```
In [3]:data
```

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi	Out
0	1	Garner Fork	Carolyn	Murray	52.0	Head	Married	Female	Copy	None	Metho	
1	1	Garner Fork	Kyle	Murray	53.0	Husband	Married	Male	Immigrati officer	None	Metho	
2	1	Garner Fork	Suzanne	Elliott- Murray	24.0	Daughter	Single	Female	Merchant navy officer	None	Metho	
3	2	Garner Fork	Naomi	Archer	34.0	Head	Single	Female	Merchant navy officer	None	None	
4	2	Garner Fork	Martin	Barnes	36.0	Partner	Single	Male	Futures trader	None	None	
...	
8873	1	Price Institute	Wendy	Farrell	38.0	Head	Single	Female	Unemplo	None	Muslim	
8874	1	Price Institute	Graham	Farrell	2.0	Son	NaN	Male	Child	None	NaN	
8875	1	Price Institute	Janet	Dean	39.0	Cousin	Single	Female	Make	None	None	
8876	1	Price Institute	Shirley	Smith	63.0	Sibling	Divorced	Female	Art gallery manager	None	Metho	
8877	1	Price Institute	Neil	Wilkinson	51.0	Lodger	Divorced	Male	Surveyor, minerals	None	Christi	

```
In [4]:# Info of the categories under the Religion coloumn
```

```
print(data['Religion'].unique())
```

Loading [MathJax]/extensions/MathZoom.js

```
['Methodist' 'None' 'Catholic' 'Christian' nan 'Muslim' 'Sikh' 'Jewish'
 'Sith' 'Bahai' 'Private' 'Nope' 'Hindu']
```

```
In [5]:# Info of the population under the Marital Status coloumn
```

```
print(data['Marital Status'].unique())
```

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

```
In [6]:# Checking for Null Values
```

```
data['Religion'].isnull().values.any()
```

Out[6]:

```
True
```

```
In [7]:# Checking for Columns with Null Values
```

```
data.isnull().sum()
```

Out[7]:

```
House Number          0
Street                0
First Name            0
Surname               0
Age                   0
Relationship to Head of House  0
Marital Status        1967
Gender                0
Occupation            0
Infirmary             0
Religion              2021
dtype: int64
```

```
In [8]:# Find rows with Marital Status
```

```
data[data['Marital Status'].isnull()]
```

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi	Outl
12	6	Garner Fork	Luke	Stone- Marsh	16.0	Son	NaN	Male	Student	None	NaN	
34	2	Brightloo Lane	Kayleigh	Jones	15.0	Daughter	NaN	Female	Student	None	NaN	
44	1	Beech Lake	Rhys	Skinner	12.0	Son	NaN	Male	Student	None	NaN	
55	1	Todd Row	Megan	Jackson	4.0	Daughter	NaN	Female	Child	None	NaN	
60	3	Todd Row	Marie	Simpson	10.0	Daughter	NaN	Female	Student	None	NaN	
...	
8857	1	Hale Strongho	Stewart	Ford	13.0	Son	NaN	Male	Student	None	NaN	
8865	1	Glover Manorho	Sarah	White	13.0	Daughter	NaN	Female	Student	None	NaN	
8866	1	Glover Manorho	Stewart	White	8.0	Son	NaN	Male	Student	None	NaN	
8867	1	Glover Manorho	Jonathan	White	7.0	Son	NaN	Male	Student	None	NaN	
8874	1	Price Institute	Graham	Farrell	2.0	Son	NaN	Male	Child	None	NaN	

```
In [9]:# replace the Marital Status Null with Single
```

```
data['Marital Status'] = data['Marital Status'].fillna('Single')
```

```
In [10]:# Find rows with Religion (Null)
```

```
data[data['Religion'].isnull()]
```

Loading [MathJax]/extensions/MathZoom.js

Out[1]



	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
12	6	Garner Fork	Luke	Stone-Marsh	16.0	Son	Single	Male	Student	None	NaN
34	2	Brightloo Lane	Kayleigh	Jones	15.0	Daughter	Single	Female	Student	None	NaN
44	1	Beech Lake	Rhys	Skinner	12.0	Son	Single	Male	Student	None	NaN
55	1	Todd Row	Megan	Jackson	4.0	Daughter	Single	Female	Child	None	NaN
60	3	Todd Row	Marie	Simpson	10.0	Daughter	Single	Female	Student	None	NaN
...
8857	1	Hale Strongho	Stewart	Ford	13.0	Son	Single	Male	Student	None	NaN
8865	1	Glover Manorho	Sarah	White	13.0	Daughter	Single	Female	Student	None	NaN
8866	1	Glover Manorho	Stewart	White	8.0	Son	Single	Male	Student	None	NaN
8867	1	Glover Manorho	Jonathan	White	7.0	Son	Single	Male	Student	None	NaN
8874	1	Price Institute	Graham	Farrell	2.0	Son	Single	Male	Child	None	NaN



In []:

In [11]:# Find rows with blank spaces

data.isin([' ']).sum()

Out[11]:

```

House Number      0
Street            0
First Name        0
Surname           2
Age              0
Relationship to Head of House  4
Marital Status    1
Gender            2
Occupation        2
Infirmity         8
Religion          0
dtype: int64

```

Loading [MathJax]/extensions/MathZoom.js

```
In [12]:# Find rows with Religion (NaN)
```

```
data[data['Religion'].isna()]
```

Out[12]:



	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
12	6	Garner Fork	Luke	Stone- Marsh	16.0	Son	Single	Male	Student	None	NaN
34	2	Brightloo Lane	Kayleigh	Jones	15.0	Daughter	Single	Female	Student	None	NaN
44	1	Beech Lake	Rhys	Skinner	12.0	Son	Single	Male	Student	None	NaN
55	1	Todd Row	Megan	Jackson	4.0	Daughter	Single	Female	Child	None	NaN
60	3	Todd Row	Marie	Simpson	10.0	Daughter	Single	Female	Student	None	NaN
...
8857	1	Hale Strongho	Stewart	Ford	13.0	Son	Single	Male	Student	None	NaN
8865	1	Glover Manorho	Sarah	White	13.0	Daughter	Single	Female	Student	None	NaN
8866	1	Glover Manorho	Stewart	White	8.0	Son	Single	Male	Student	None	NaN
8867	1	Glover Manorho	Jonathan	White	7.0	Son	Single	Male	Student	None	NaN
8874	1	Price Institute	Graham	Farrell	2.0	Son	Single	Male	Child	None	NaN



```
In [13]:# replace the Religion blank space with None
```

```
data['Religion'] = data['Religion'].fillna('None')
```

```
In [14]:print(data['Religion'].unique())
```

```
['Methodist' 'None' 'Catholic' 'Christian' 'Muslim' 'Sikh' 'Jewish' 'Sith'
 'Bahai' 'Private' 'Nope' 'Hindu']
```

```
In [15]:# replace the Religion with Private and Nope with None
```

```
data['Religion'].replace(
    {'Private':'None', 'Nope':'None'}, inplace = True)
```

```
In [16]:print(data['Religion'].unique())
```

Loading [MathJax]/extensions/MathZoom.js

```
['Methodist' 'None' 'Catholic' 'Christian' 'Muslim' 'Sikh' 'Jewish' 'Sith'
 'Bahai' 'Hindu']
```

```
In [17]:#convert the Ages float to Int
```

```
data['Age'] = data['Age'].astype(int)
```

```
In [18]:data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 8878 entries, 0 to 8877
```

```
Data columns (total 11 columns):
```

#	Column	Non-Null Count	Dtype
0	House Number	8878 non-null	int64
1	Street	8878 non-null	object
2	First Name	8878 non-null	object
3	Surname	8878 non-null	object
4	Age	8878 non-null	int32
5	Relationship to Head of House	8878 non-null	object
6	Marital Status	8878 non-null	object
7	Gender	8878 non-null	object
8	Occupation	8878 non-null	object
9	Infirmity	8878 non-null	object
10	Religion	8878 non-null	object

```
dtypes: int32(1), int64(1), object(9)
```

```
memory usage: 728.4+ KB
```

```
In [19]:# Print the unique Ages
```

```
print(data['Age'].unique())
```

```
[ 52  53  24  34  36  71  28  30  82  48  51  16  45  50  25  70  75  66
  35  33  23  74  60  62  38  69  21  46  15  54  73  26  44  12  61  19
  18  58   4  43  47  10   5   1  41  27  72  78  77  37  14   9  40   6
  56   8  68  67  63  64  83  29  32  22  13  55  57  81  59  85  86  11
   7  39  20   0  31   2  17   3  49  42  79  80  84  76  65  89 105  88
  90  95 103  96  93  87  92  94 100  98  91 104 102 101  99]
```

```
In [20]:data['Age']
```

Out[20]:

```
0      52
1      53
2      24
3      34
4      36
..
8873   38
8874    2
8875   39
8876   63
8877   51
```

```
Name: Age, Length: 8878, dtype: int32
```

```
In [21]:# Finding rows with with Empty value and putting replacements
```

```
In [22]:# Find rows with Empty values in Surname
```

```
data[data['Surname'].isin([' '])]
```

Out[2]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
1418	7	Myers Corners	Naomi		25	Lodger	Single	Female	Press photogra	None	None
4515	2	Swansea	William		36	None	Single	Male	Secretary,	None	Christi

```
In [23]:# Find info about people living in 7 Myers Corners
```

```
data.loc[(data['House Number'] == 7) & (data['Street'] == 'Myers Corners')]
```

Out[2]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
1417	7	Myers Corners	Tina	Ellis	30	Head	Divorced	Female	Insurance underwrit	None	None
1418	7	Myers Corners	Naomi		25	Lodger	Single	Female	Press photogra	None	None
1419	7	Myers Corners	Allan	Williams	41	Lodger	Single	Male	Diagnosti radiograp	None	Christi
1420	7	Myers Corners	Roy	Graham	47	Lodger	Single	Male	Tax adviser	None	None
1421	7	Myers Corners	Jenna	Ahmed	57	Lodger	Single	Female	Mechanic engineer	None	Christi
1422	7	Myers Corners	Tracey	Bailey	62	Lodger	Single	Female	Trade union research	None	None

```
In [24]:# Replace the blank Surname with 'Unknown'
```

```
data.at[1418, 'Surname'] = 'Unknown'
```

```
In [25]:# Find info about people living in 2 Swansea Stream
```

```
data.loc[(data['House Number'] == 2) & (data['Street'] == 'Swansea Stream')]
```

Loading [MathJax]/extensions/MathZoom.js

Out[2]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
4514	2	Swansea Stream	Kirsty	Harding	26	Head	Single	Female	Sports coach	None	None
4515	2	Swansea Stream	William		36	None	Single	Male	Secretary, company	None	Christi
4516	2	Swansea Stream	Janice	Carr	28	None	Single	Female	Surveyor, hydrograj	None	None
4517	2	Swansea Stream	Kyle	Mellor	54	None	Divorced	Male	Retail manager	None	Christi
4518	2	Swansea	Timothv	Harrison	37	None	Single	Male	Unemplo	None	Christi

In [26]:# Replace the blank Surname with 'Unknown'

```
data.at[4515, 'Surname' ]='Unknown'
```

In [27]:# Find rows with Marital Status (NaN)

```
data[data['Marital Status'].isna()]
```

Out[2]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
--	-----------------	--------	---------------	---------	-----	--	-------------------	--------	----------	-----------	--------

In [28]:# replace the Marital Status blank space with Single

```
data['Marital Status'] = data['Marital Status'].fillna('Single')
```

In [29]:# replace the Marital Status blank space for children less than 18 by NA

```
data[(data['Age']>=0) & (data['Age']<=17)]
```

Loading [MathJax]/extensions/MathZoom.js

Out[2]



	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
12	6	Garner Fork	Luke	Stone- Marsh	16	Son	Single	Male	Student	None	None
34	2	Brightloo Lane	Kayleigh	Jones	15	Daughter	Single	Female	Student	None	None
44	1	Beech Lake	Rhys	Skinner	12	Son	Single	Male	Student	None	None
55	1	Todd Row	Megan	Jackson	4	Daughter	Single	Female	Child	None	None
60	3	Todd Row	Marie	Simpson	10	Daughter	Single	Female	Student	None	None
...
8857	1	Hale Strongho	Stewart	Ford	13	Son	Single	Male	Student	None	None
8865	1	Glover Manorho	Sarah	White	13	Daughter	Single	Female	Student	None	None
8866	1	Glover Manorho	Stewart	White	8	Son	Single	Male	Student	None	None
8867	1	Glover Manorho	Jonathan	White	7	Son	Single	Male	Student	None	None
8874	1	Price Institute	Graham	Farrell	2	Son	Single	Male	Child	None	None



```
In [30]: minors = ((data['Age'] >= 0) & (data['Age'] <= 17))
         minors
```

Out[30]:

```
0      False
1      False
2      False
3      False
4      False
...
8873   False
8874    True
8875   False
8876   False
8877   False
```

```
Name: Age, Length: 8878, dtype: bool
```

```
In [31]: data.loc[minors, 'Marital Status'] = 'NA'
Loading [MathJax]/extensions/MathZoom.js
In [32]: # Print info on Population with Age 0-17
```

```
data[(data['Age']>=0) & (data['Age']<=17)]
```

Out[3]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
12	6	Garner Fork	Luke	Stone- Marsh	16	Son	NA	Male	Student	None	None
34	2	Brightloo Lane	Kayleigh	Jones	15	Daughter	NA	Female	Student	None	None
44	1	Beech Lake	Rhys	Skinner	12	Son	NA	Male	Student	None	None
55	1	Todd Row	Megan	Jackson	4	Daughter	NA	Female	Child	None	None
60	3	Todd Row	Marie	Simpson	10	Daughter	NA	Female	Student	None	None
...
8857	1	Hale Strongho	Stewart	Ford	13	Son	NA	Male	Student	None	None
8865	1	Glover Manorho	Sarah	White	13	Daughter	NA	Female	Student	None	None
8866	1	Glover Manorho	Stewart	White	8	Son	NA	Male	Student	None	None
8867	1	Glover Manorho	Jonathan	White	7	Son	NA	Male	Student	None	None
8874	1	Price Institute	Graham	Farrell	2	Son	NA	Male	Child	None	None

```
In [33]:# Info of the categories under the Marital Stautus coloumn
```

```
print(data['Marital Status'].unique())
```

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

```
In [34]:# Find rows with empty Relationship to Head of House
```

```
data[data['Relationship to Head of House'].isin([' '])]
```

Loading [MathJax]/extensions/MathZoom.js

Out[3]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
684	179	Baker Branch	Nathan	Dunn	69		Married	Male	Retired Psycholog prison and probation ser...	None	Christi
1118	84	Mary Center	Caroline	Brown	56		Single	Female	Health visitor	None	Christi
2069	2	Calendar Coves	Holly	Mills	19		Single	Female	University Student	None	Metho
4234	3	Lawrence	Vincent	Ward	14		NA	Male	Student	None	None

In [35]:# Find info about people living in 179 Baker Branch

```
data.loc[(data['House Number'] == 179) & (data['Street'] == 'Baker Branch')]
```

Out[3]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
684	179	Baker Branch	Nathan	Dunn	69		Married	Male	Retired Psycholog prison and probation ser...	None	Christi
685	179	Baker	Francesca	Dunn	66	Wife	Married	Female	Health	None	Christi

In [36]:# Replace the blank Relationship to Head of House with 'Head'

```
data.at[684, 'Relationship to Head of House'] = 'Head'
```

In [37]:# Find info about people living in 84 Mary Center

```
data.loc[(data['House Number'] == 84) & (data['Street'] == 'Mary Center')]
```

Loading [MathJax]/extensions/MathZoom.js

Out[3]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
1116	84	Mary Center	Katy	Craig	37	Head	Single	Female	Doctor, general practice	None	Cathol
1117	84	Mary Center	Anna	Craig	0	Daughter	NA	Female	Child	None	None
1118	84	Mary	Caroline	Brown	56		Single	Female	Health	None	Christi

```
In [38]:# Replace the blank Relationship to Head of House with 'Lodger'
```

```
data.at[1118,'Relationship to Head of House' ]='Lodger'
```

```
In [39]:# Find info about people living in 2 Calendar Coves
```

```
data.loc[(data['House Number'] == 2) & (data['Street'] == 'Calendar Coves')]
```

Out[3]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
2069	2	Calendar	Hollv	Mills	19		Single	Female	University	None	Metho

```
In [40]:# Replace the blank Relationship to Head of House with 'Mother'
```

```
data.at[2069,'Relationship to Head of House' ]='Head'
```

```
In [41]:# Find info about people living in 3 Lawrence Canyon
```

```
data.loc[(data['House Number'] == 3) & (data['Street'] == 'Lawrence Canyon')]
```

Out[4]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
4232	3	Lawrence Canyon	Louise	Ward	42	Head	Married	Female	Engineeri geologist	None	Cathol
4233	3	Lawrence Canyon	Joyce	Ward	41	Wife	Married	Female	Public relations account executive	None	None
4234	3	Lawrence	Vincent	Ward	14		NA	Male	Student	None	None

```
In [42]:# Replace the blank Relationship to Head of House with 'Child'
```

```
data.at[4234, 'Relationship to Head of House' ]='Child'
In [43]:# Find rows with empty Marital Status
```

```
data[data['Marital Status'].isin([' '])]
```

Out[4]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
1966	57	Tamarisk Road	James	Parker	54	Husband		Male	Surveyor, planning and	None	None

```
In [44]:# Replace the blank Marital Status with 'Married'
```

```
data.at[1966, 'Marital Status' ]='Married'
```

```
In [45]:# Find rows with empty Gender
```

```
data[data['Gender'].isin([' '])]
```

Out[4]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
5059	14	Bright Avenue	Jay	Sharpe	6	Son	NA		Student	None	None
5164	49	Bright	Mohamm	Williams	20	Son	Divorced		University	None	None

```
In [46]:# Replace the Gender with 'Male' due to the relationship to the head of hou
```

```
data.at[5059, 'Gender' ]='Male'
```

```
In [47]:# Replace the blank Gender with 'Male'
```

```
data.at[5164, 'Gender' ]='Male'
```

```
In [48]:# Find rows with empty Occupation
```

```
data[data['Occupation'].isin([' '])]
```

Out[4]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
5662	133	Edinburgl Points	Lesley	Green	71	Head	Married	Female		None	Christi
8519	13	Morgan	Steven	Norman	46	Partner	Single	Male		None	None

```
In [49]:# Replace the blank Gender with 'Retired'
```

```
data.at[5662, 'Occupation' ]='Retired'
```

```
In [50]:# Find info about people living in 13 Morgan Crescent
```

```
data.loc[(data['House Number'] == 13) & (data['Street'] == 'Morgan Crescent
```

Out[5]

	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
8518	13	Morgan Crescent	Abbie	Potts	39	Head	Single	Female	Materials engineer	None	None
8519	13	Morgan Crescent	Steven	Norman	46	Partner	Single	Male		None	None
8520	13	Morgan Crescent	Lee	Potts	13	Son	NA	Male	Student	None	None
8521	13	Morgan	Ann	Potts	11	Daughter	NA	Female	Student	None	None

```
In [51]:# Replace the blank Occupation with 'Unemployed'
```

```
data.at[8519, 'Occupation' ]='Unemployed'
```

```
In [52]:# Show the Unique Occupation
```

```
print(data['Occupation'].unique())
```

```
['Copy' 'Immigration officer' 'Merchant navy officer' ...
'Retired Transport planner' 'Retired Multimedia programmer'
'Retired Animal nutritionist']
```

```
In [53]:data['Occupation']
```

Out[53]:

```

0          Copy
1  Immigration officer
2  Merchant navy officer
3  Merchant navy officer
4      Futures trader
...
8873      Unemployed
8874      Child
8875      Make
8876  Art gallery manager
8877  Surveyor, minerals
Name: Occupation, Length: 8878, dtype: object
In [54]:# Info on the Unemployed Population

print(data['Occupation']== 'Unemployed')

0      False
1      False
2      False
3      False
4      False
...
8873     True
8874     False
8875     False
8876     False
8877     False
Name: Occupation, Length: 8878, dtype: bool
In [55]:# Find rows with empty Infirmary

data[data['Infirmary'].isin([' '])]

```

Loading [MathJax]/extensions/MathZoom.js

Out[5]



	House Number	Street	First Name	Surname	Age	Relations to Head of House	Marital Status	Gender	Occupati	Infirmity	Religi
3690	7	Marshall Inlet	Thomas	Wright	38	Husband	Married	Male	Firefighte		None
3879	7	Station Road	Kayleigh	Taylor	40	Head	Married	Female	Teacher, English as a foreign language		Christi
5891	13	James Station	Samantha	Hewitt	55	Head	Single	Female	Public affairs consultan		Cathol
6296	1	Sydney Drive	Melanie	Brown	27	Head	Single	Female	Applicatio develop		None
6772	34	Summer Villages	Gareth	Phillips	57	Head	Married	Male	Psycholog counsellir		Metho
7672	18	Begum Ridges	Carly	Waters	15	Daughter	NA	Female	Student		None
7780	2	Coconut Lane	Ross	Newton	61	Head	Single	Male	Newspap journalist		Christi
8732	1	Ward Fields	Francesca	Edwards	32	Head	Married	Female	Historic buildings inspector,		None



```
In [56]:# Replace the blank Infirmity with 'None'
```

```
data['Infirmity'] = data['Infirmity'].fillna('None')
```

DATA VISUALIZATION

Plotting the AGE PYRAMID in the Census Data

```
In [57]:# male age data info
```

```
filt = data['Gender'] == 'Male'
male_data = data.loc[filt, 'Age']
male_data
```

Loading [MathJax]/extensions/MathZoom.js

Out[57]:

```

1      53
4      36
7      30
9      82
11     51
      ..
8867    7
8869    32
8872    39
8874     2
8877    51

```

Name: Age, Length: 4234, dtype: int32

In [58]:male_data.values

Out[58]:

```
array([53, 36, 30, ..., 39,  2, 51])
```

In [59]:# Female age data info

```

filt = data['Gender'] == 'Female'
female_data = data.loc[filt, 'Age']
female_data

```

Out[59]:

```

0      52
2      24
3      34
5      71
6      28

```

```

      ..
8870    27
8871    63
8873    38
8875    39
8876    63

```

Name: Age, Length: 4644, dtype: int32

In [60]:# Create the age groups for the x-axis

```

age_bins = {'0-4' : 0, '5-9' : 0, '10-14' : 0, '15-19' : 0, '20-24' : 0, '2
          '30-34' : 0 , '35-39' : 0 , '40-44' : 0, '45-49' : 0 , '50-54
          '60-64' : 0, '65-69' : 0, '70-74' : 0, '75-79' : 0, '80-84' :

```

In [61]:def create_bins(age,age_bin):

```
    bins = age_bins.copy()
```

```
    age = age.values
```

```
    for a in age:
```

```
        if 0<a<4:
```

```
            bins['0-4']+=1
```

```
        if 5<a<9:
```

```
            bins['5-9']+=1
```

```
        if 10<a<14:
```

```
            bins['10-14']+=1
```

```
        if 15<a<19:
```

```
            bins['15-19']+=1
```

```
        if 20<a<24:
```

Loading [MathJax]/extensions/MathZoom.js

```

        bins['20-24']+=1
    if 25<a<29:
        bins['25-29']+=1
    if 30<a<34:
        bins['30-34']+=1
    if 35<a<39:
        bins['35-39']+=1
    if 40<a<44:
        bins['40-44']+=1
    if 45<a<49:
        bins['45-49']+=1
    if 50<a<54:
        bins['50-54']+=1
    if 55<a<59:
        bins['55-59']+=1
    if 60<a<64:
        bins['60-64']+=1
    if 65<a<69:
        bins['65-69']+=1
    if 70<a<74:
        bins['70-74']+=1
    if 75<a<79:
        bins['75-79']+=1
    if 80<a<84:
        bins['80-84']+=1
    if 85<a<89:
        bins['85-89']+=1
    if 90<a<94:
        bins['90-94']+=1
    if 95<a<99:
        bins['95-99']+=1
    if 100<a<104:
        bins['100-104']+=1
    return bins

```

```

In [62]:m_age_bins = create_bins(male_data,age_bins)
        m_age_count = [-1*age for age in m_age_bins.values()]
        m_age_count.reverse()
        male_age = list(m_age_bins.keys())
        male_age.reverse()

```

```

In [63]:f_age_bins = create_bins(female_data,age_bins)
        f_age_count = [age for age in f_age_bins.values()]
        f_age_count.reverse()
        female_age = list(f_age_bins.keys())
        female_age.reverse()

```

```

In [64]:# Create the pyramid plot

```

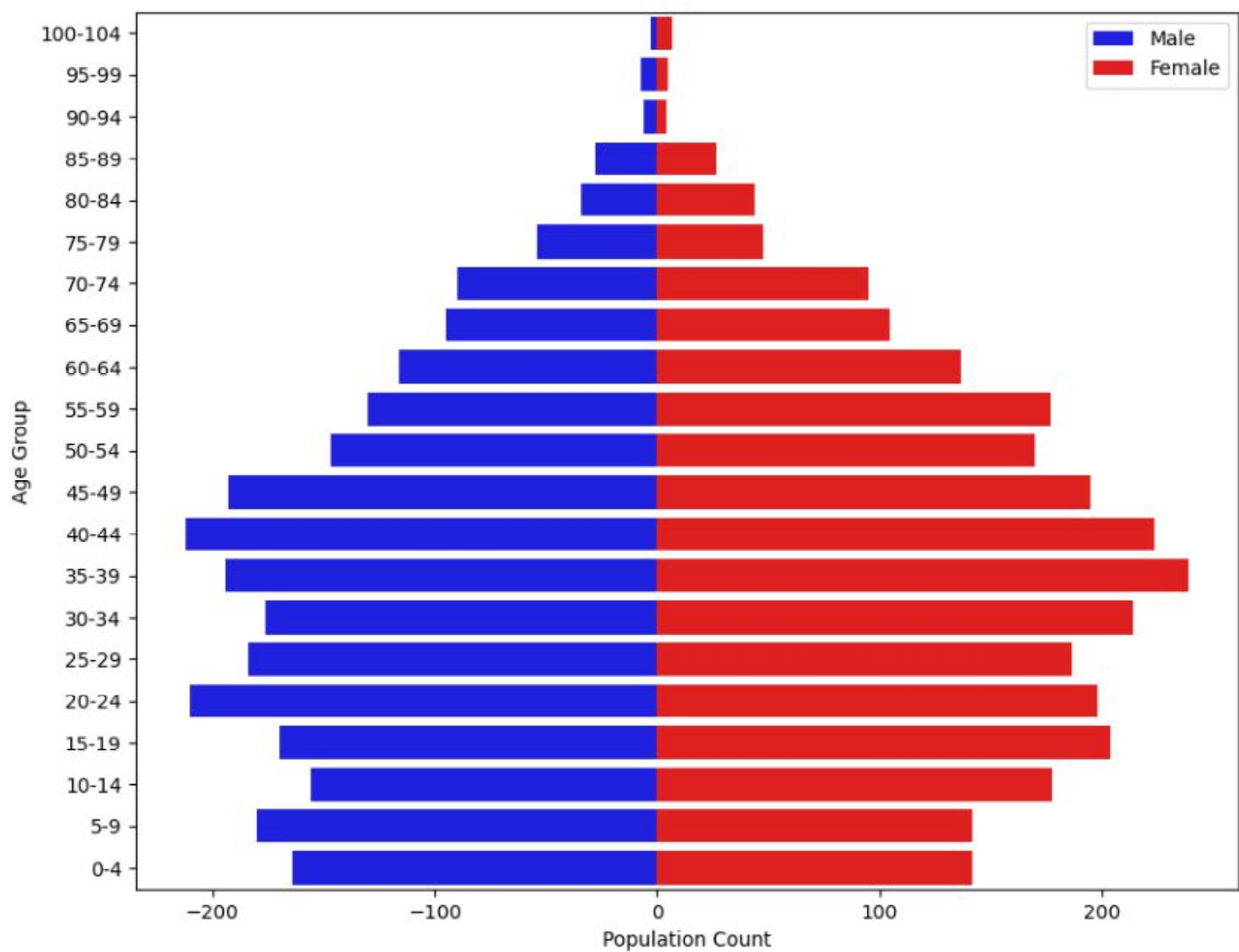
```

sns.barplot(x = m_age_count,y = male_age,color = 'blue', label = 'Male')
sns.barplot(x = f_age_count,y = female_age,color = 'red', label = 'Female')
plt.legend()
plt.xlabel('Population Count')
plt.ylabel('Age Group')
fig = plt.gcf()

```

Loading [MathJax]/extensions/MathZoom.js

```
fig.set_size_inches(10,8)
```



from the Age pyramid above, we can see that there is a high population of residents of the town from age bracket 20 - 49 who are basically university students and working class people. There will also be a lot of commuting in and out of the town for studies and work thereby making this age groups commuters in the town. Therefore, building a train station in on the unused land will be beneficial to the community as there is also a growing population of children in the town.

RELIGIOUS AFFILIATIONS

In [65]:# Create the Religion plot

```
Religionbins=sns.displot(data, x='Religion', bins=10)
```

```
# Rotate x-axis labels by 90 degrees
```

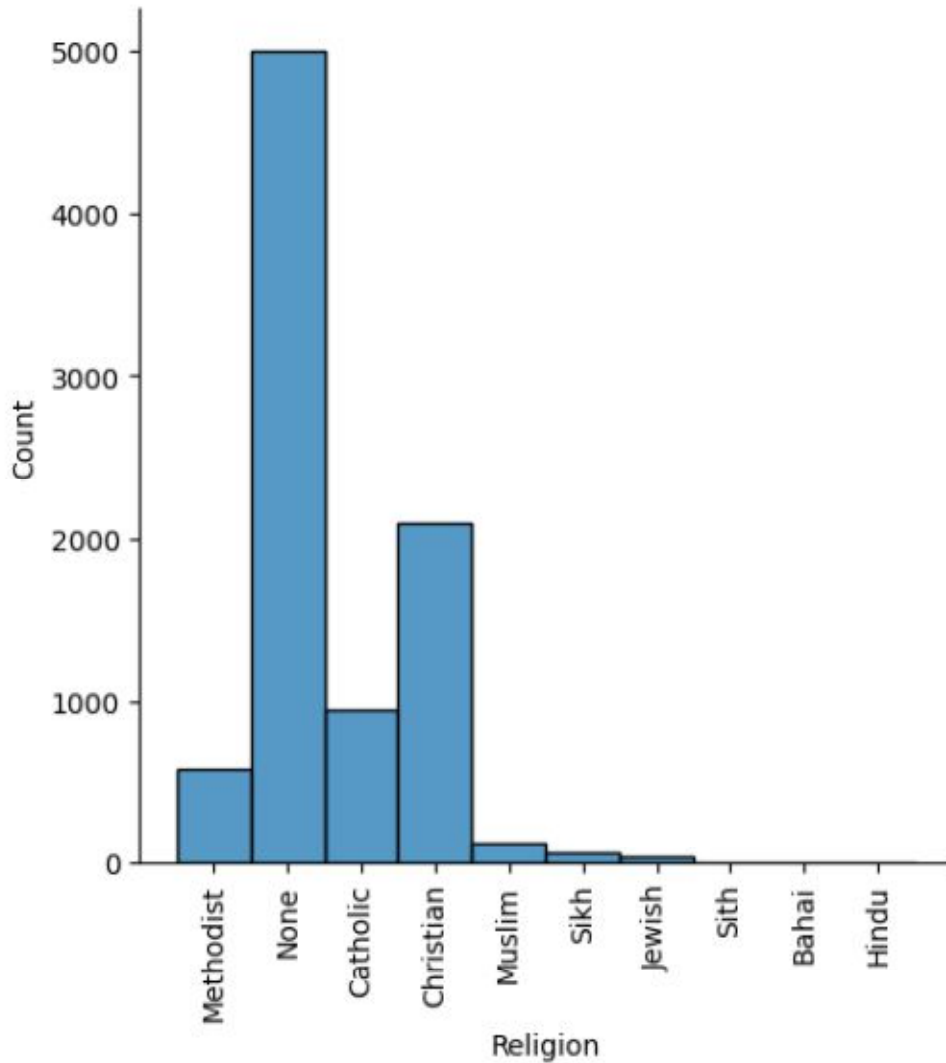
```
for item in Religionbins.ax.get_xticklabels():
    item.set_rotation(90)
```

```
# Add a descriptive label to the x-axis
```

```
Religionbins.set_axis_labels('Religion', 'Count')
```

```
# Show the plot
```

```
plt.show()
```



DIVORCE RATE AND MARRIAGE RATE

In [66]:# Create the Marital Status plot

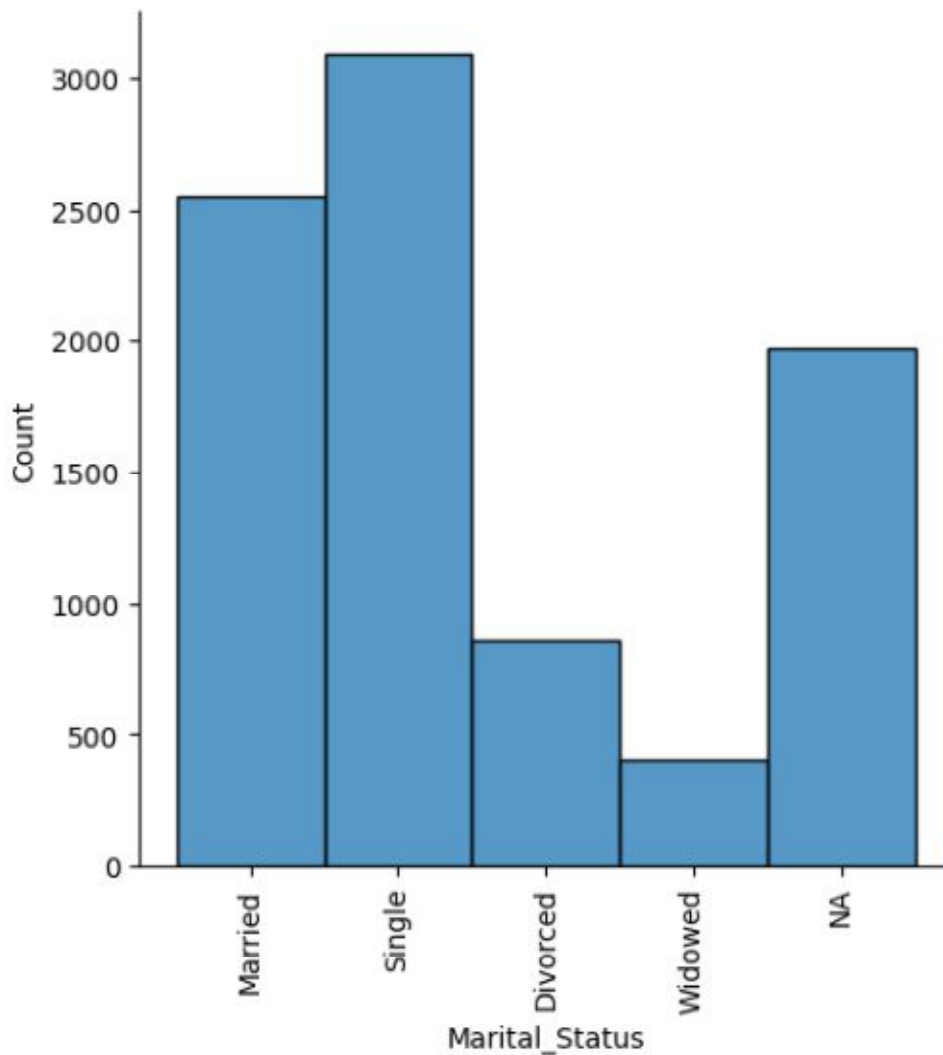
```
Marital_Statusbins=sns.displot(data, x='Marital Status', bins=10)

# Rotate x-axis labels by 90 degrees
for item in Marital_Statusbins.ax.get_xticklabels():
    item.set_rotation(90)

# Add a descriptive label to the x-axis
Marital_Statusbins.set_axis_labels('Marital Status', 'Count')

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

Loading [MathJax]/extensions/MathZoom.js



From the above histogram, we can see that the Marriage rate is higher than the Divorce rate. Furthermore, the histogram shows that there is a dense population of Singles in the town and some of them will most likely change their status to Married in the nearest future. Considering this analysis, it is evident that there will be a growing population of school children from the married couples, new births due to new marriages, therefore I will suggest that the schooling budget should increase.

UNEMPLOYMENT TRENDS

In [67]:# Info on the categories of Occupation

```
data['Occupation'].unique()
```

Out[67]:

```
array(['Copy', 'Immigration officer', 'Merchant navy officer', ...,
      'Retired Transport planner', 'Retired Multimedia programmer',
      'Retired Animal nutritionist'], dtype=object)
```

In [68]:data['Occupation'].value_counts()

Out[68]:

```
Student          1577
University Student  592
Unemployed       590
Child            494
PhD Student      20
...
Retired Designer, interior/spatial  1
Retired Gaffer                     1
Retired Sports development officer  1
Retired Public relations officer   1
Retired Animal nutritionist        1
Name: Occupation, Length: 1126, dtype: int64
In [69]:# Checking for Variations in Occupation by Age
```

```
data[['Age', 'Occupation']].groupby('Occupation').describe()
```

Out[6

	Age							
	count	mean	std	min	25%	50%	75%	max
Occupation								
Academic librarian	8.0	46.500000	15.371588	23.0	35.50	48.0	57.00	67.0
Accommodation manager	6.0	37.833333	9.867455	26.0	30.50	37.5	44.50	51.0
Accountant, chartered	9.0	49.111111	12.323194	29.0	46.00	49.0	58.00	67.0
Accountant, chartered certified	6.0	44.333333	13.002564	32.0	34.25	42.0	49.75	66.0
Accountant, chartered management	8.0	45.000000	8.585702	32.0	41.50	46.0	49.50	58.0
...
Water quality scientist	6.0	50.000000	10.119289	35.0	43.25	53.0	56.75	61.0
Web designer	6.0	45.166667	13.392784	31.0	35.25	42.0	54.00	65.0
Wellsite geologist	7.0	40.000000	9.327379	26.0	34.50	40.0	46.50	52.0
Writer	7.0	48.142857	16.035675	29.0	33.00	53.0	62.00	65.0
Youth worker	11.0	44.636364	8.164224	32.0	39.50	46.0	50.50	54.0

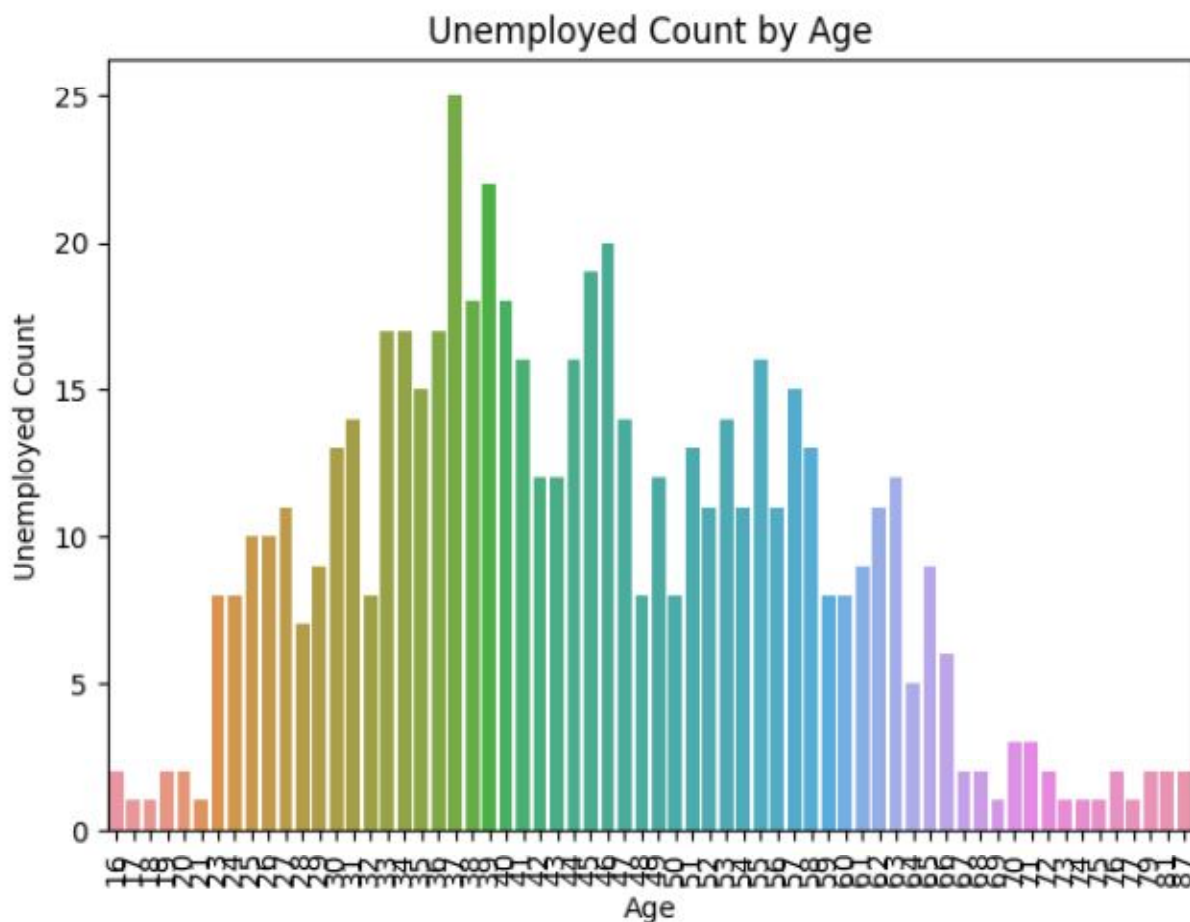
Loading [MathJax]/extensions/MathZoom.js

In [70]:# Unemployed Count by Age Plot

```
Unemployed_Population = data[data['Occupation'] == 'Unemployed']
In [71]:plt.figure(figsize=(7,5))
Unemployed_plot = sns.countplot(data = Unemployed_Population, x = 'Age' )
# Rotate x-axis labels by 90 degrees
for item in Unemployed_plot.get_xticklabels():
    item.set_rotation(90)
Unemployed_plot.set_title('Unemployed Count by Age')
Unemployed_plot.set_ylabel('Unemployed Count')

Text(0, 0.5, 'Unemployed Count')
```

Out[71]:



RELIGIOUS AFFILIATIONS BY AGE

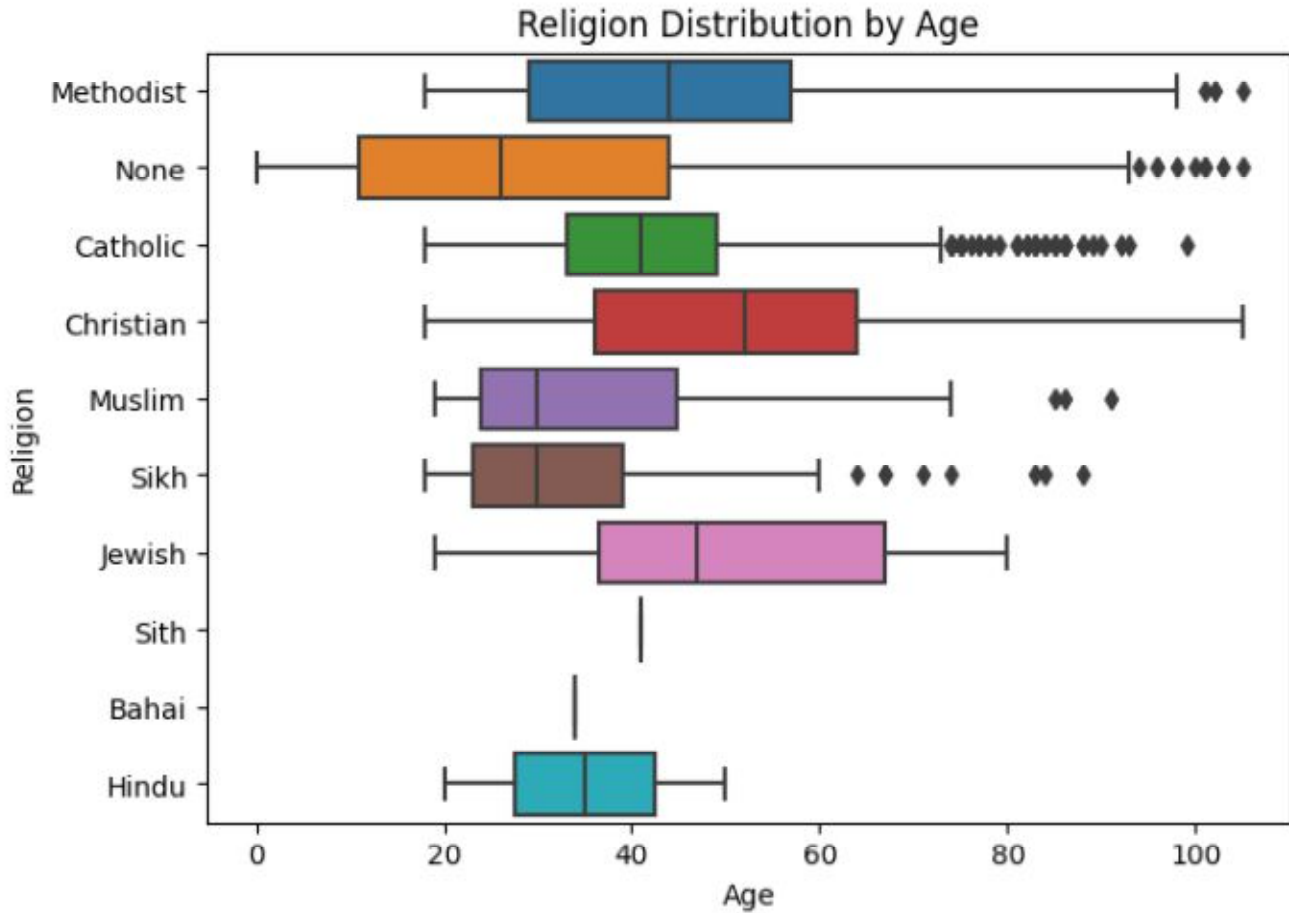
In [72]:# Checking for Religion Distribution by Age

```
plt.figure(figsize=(7,5))
Religion_plot = sns.boxplot(y = 'Religion', x = 'Age', data = data)
Religion_plot.set_title ('Religion Distribution by Age')
```

Loading [MathJax]/extensions/MathZoom.js

Out[72]:

```
Text(0.5, 1.0, 'Religion Distribution by Age')
```



```
In [73]:# Checking for Variations in Religion by Age
```

```
data[['Age', 'Religion']].groupby('Religion').describe()
```

Loading [MathJax]/extensions/MathZoom.js

Out[7]

	Age							
	count	mean	std	min	25%	50%	75%	max
Religion								
Bahai	2.0	34.000000	0.000000	34.0	34.0	34.0	34.00	34.0
Catholic	955.0	42.556021	15.273322	18.0	33.0	41.0	49.00	99.0
Christian	2100.0	50.781905	18.587420	18.0	36.0	52.0	64.00	105.0
Hindu	2.0	35.000000	21.213203	20.0	27.5	35.0	42.50	50.0
Jewish	40.0	50.200000	17.221334	19.0	36.5	47.0	67.00	80.0
Methodist	578.0	44.519031	17.880127	18.0	29.0	44.0	57.00	105.0
Muslim	130.0	36.600000	16.699626	19.0	24.0	30.0	44.75	91.0
None	5005.0	29.395005	21.543016	0.0	11.0	26.0	44.00	105.0
Sikh	65.0	35.815385	18.109815	18.0	23.0	30.0	39.00	88.0

MARRIAGE RATE AND DIVORCE RATE

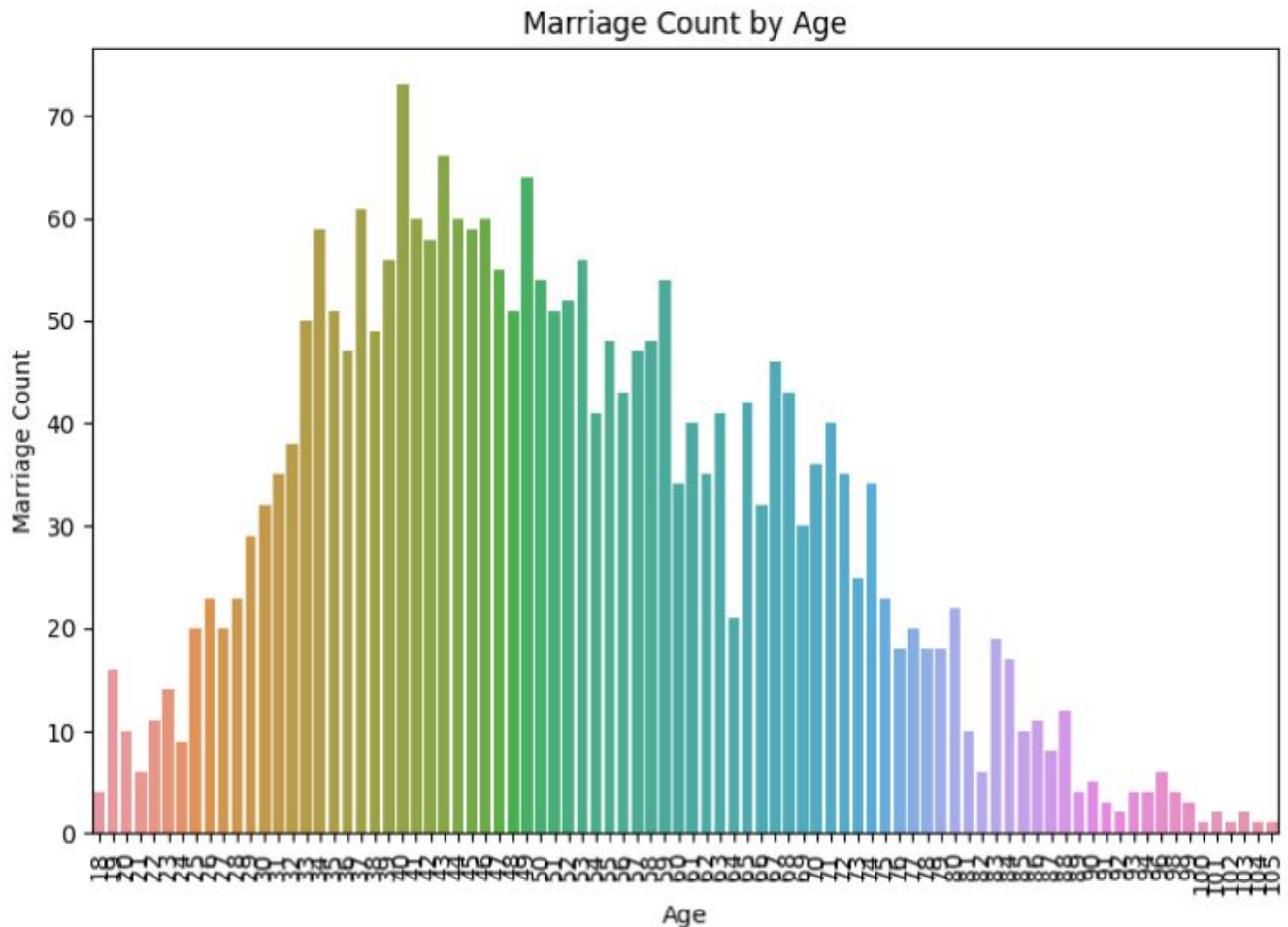
In [74]:# *Marriage Count Plot by Age*

```
Married_Population = data[data['Marital Status'] == 'Married']
In [75]:plt.figure(figsize=(9,6))
Married_plot = sns.countplot(data = Married_Population, x = 'Age' )
# Rotate x-axis labels by 90 degrees
for item in Married_plot.get_xticklabels():
    item.set_rotation(90)
Married_plot.set_title('Marriage Count by Age')
Married_plot.set_ylabel('Marriage Count')
```

Loading [MathJax]/extensions/MathZoom.js

Out[75]:

```
Text(0, 0.5, 'Marriage Count')
```



```
In [76]:# Checking for Variations in Marital Status by Age
```

```
data[['Age', 'Marital Status']].groupby('Marital Status').describe()
```

Out[7]

	Age							
	count	mean	std	min	25%	50%	75%	max
Marital Status								
Divorced	860.0	41.943023	16.575794	18.0	29.0	39.0	53.0	89.0
Married	2552.0	51.704154	16.724315	18.0	39.0	50.0	64.0	105.0
NA	1970.0	8.879695	5.106310	0.0	4.0	9.0	13.0	17.0
Single	3093.0	37.720983	14.339199	18.0	25.0	36.0	48.0	100.0

```
In [77]:# Percentage of Married Population
```

```
Loading [MathJax]/extensions/MathZoom.js
```

```
Percentage_Married_Rate = round(len(Married_Population) * 100 / len(data), 2)
```

```
print(f'{Percentage_Married_Rate}%')
```

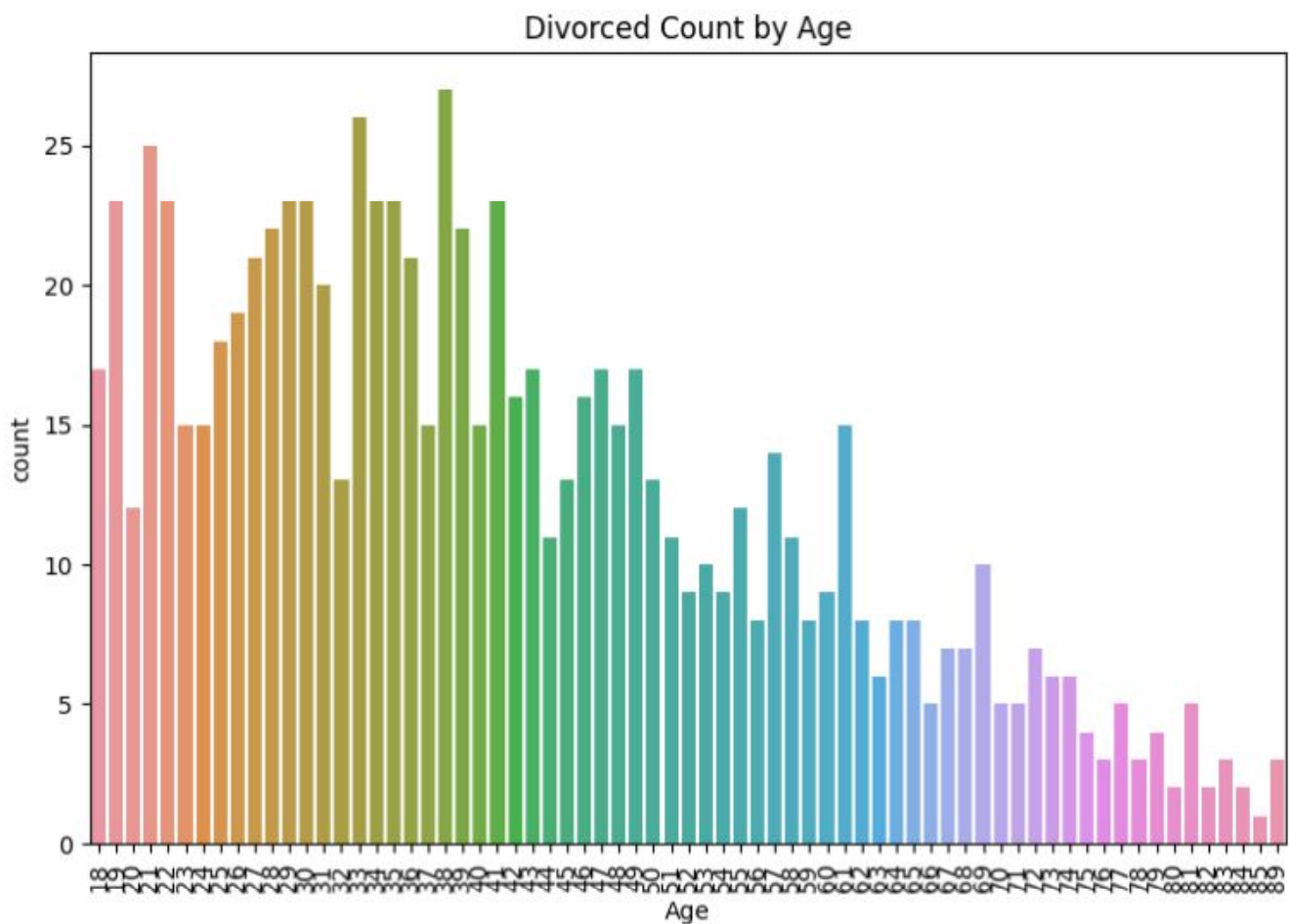
28.75%

```
In [78]:# Dovorce Count Plot by Age
```

```
Divorced_Population = data[data['Marital Status'] == 'Divorced']
In [79]:plt.figure(figsize=(9,6))
Divorced_plot = sns.countplot(data = Divorced_Population, x = 'Age' )
# Rotate x-axis labels by 90 degrees
for item in Divorced_plot.get_xticklabels():
    item.set_rotation(90)
Divorced_plot.set_title('Divorced Count by Age')
```

Out[79]:

```
Text(0.5, 1.0, 'Divorced Count by Age')
```



```
In [80]:# Percentage of Divorced Population
```

```
Percentage_Divorced_Rate = round(len(Divorced_Population)* 100/ len(data),2)
print(f'{Percentage_Divorced_Rate}%')
```

9.69%

OCCUPANCY LEVEL

```
In [81]:# Checking for Occupancy level in the Census data by Age
```

```
Occupancy = data.groupby(['Street', 'House Number']).size().reset_index(name=
Occupancy['Number of Occupants'].value_counts())
```

Out[81]:

```
2      966
1      964
3      538
4      511
5      293
6       67
7       12
12       9
8         6
10        5
11        4
15        3
9         3
13        1
21        1
17        1
```

```
Name: Number of Occupants, dtype: int64
```

BIRTH RATE

```
In [82]:# Checking for Birth rate
```

```
Babies = data[data['Age'] == 0]
Number_of_babies = len(Babies)
```

```
In [83]:Crude_birth_rate=Number_of_babies * 1000/ len(data)
Crude_birth_rate
```

Out[83]:

```
9.461590448299166
```

```
In [84]:# EVOLVING BIRTH RATE FOR 5 YEARS
```

```
Age_4 = sum(data['Age'] == 4)
Evolving_birth_rate = Age_4 * 1000/len(data)
Evolving_birth_rate
```

Out[84]:

```
11.714350078846588
```

DEATH RATE

```
In [85]:# Checking for Death rate by Age and Gender
```

```
grouped_data = data.groupby(['Age', 'Gender']).size().reset_index(name='Pop
In [86]:female_table = grouped_data [grouped_data['Gender'] == 'Female']
male_table= grouped_data[grouped_data['Gender'] == 'Male']
female_table.columns = ['Age', 'Gender', 'Female Population']
male_table.columns = ['Age', 'Gender', 'Male Population']
In [87]:Pyramid_table = pd.merge(female_table,male_table, on='Age', how='outer')
In [88]:Pyramid_table
```

Loading [MathJax]/extensions/MathZoom.js

Out[8]

	Age	Gender_x	Female Population	Gender_y	Male Population
0	0	Female	53	Male	31.0
1	1	Female	45	Male	52.0
2	2	Female	51	Male	51.0
3	3	Female	46	Male	61.0
4	4	Female	57	Male	47.0
...
100	101	Female	2	Male	2.0
101	102	Female	2	Male	1.0
102	103	Female	3	NaN	NaN
103	104	Female	1	Male	1.0
104	105	Female	1	Male	2.0

INFIRMITY

In [89]:# Checking for Infomation on the Infirmity in the Census data

```
data['Infirmity'].unique()
```

Out[89]:

```
array(['None', 'Physical Disability', 'Blind', 'Disabled',
      'Mental Disability', 'Deaf', 'Unknown Infection', ' '],
      dtype=object)
```

In [90]:data['Infirmity'].value_counts()

Out[90]:

```
None      8805
Physical Disability    18
Disabled      11
Mental Disability    10
Deaf           9
Unknown Infection     9
Blind           8
              8
```

Name: Infirmity, dtype: int64

In [91]:# Checking for Variations in Infirmity by Age

```
data[['Age', 'Infirmity']].groupby('Infirmity').describe()
```

Loading [MathJax]/extensions/MathZoom.js

								Out[9]
	Age							
	count	mean	std	min	25%	50%	75%	max
Infirmity								
	8.0	40.625000	16.097360	15.0	30.75	39.0	55.50	61.0
Blind	8.0	34.125000	19.650064	3.0	25.50	32.0	47.50	62.0
Deaf	9.0	38.666667	34.803017	2.0	13.00	20.0	66.00	94.0
Disabled	11.0	31.636364	20.857961	7.0	20.00	27.0	41.00	79.0
Mental Disability	10.0	44.400000	23.557023	3.0	32.50	39.5	56.25	88.0
None	8805.0	37.073367	21.966670	0.0	19.00	36.0	53.00	105.0
Physical Disability	18.0	46.055556	17.027563	5.0	37.75	46.0	53.00	74.0
Unknown	9.0	45.777778	12.132372	26.0	40.00	43.0	57.00	65.0

In []:

Loading [MathJax]/extensions/MathZoom.js