CENSUS PROJECT REPORT

Fundamentals of Data Science

By Azeezat Busari, 202172030

INTRODUCTION

This report showcases the analysis of a mock census of an imaginary modest town as well as insights drawn after the analysis to make some decisions on the development of the town. In order to make recommendations on what to invest in to further develop the town, the census data was cleaned thoroughly so as to ensure that the data was of good quality; after which it was adequately analyzed.

The Problem Statements are defined below:

1. ***What should be built on an unoccupied plot of land that the local government wishes to develop?***
    a. High-density housing. This should be built if the population is significantly expanding.
    b. Low-density housing. This should be built if the population is "affluent" and there is demand for large-family housing.
    c. Train station. There are potentially a lot of commuters in the town and building a train station could take pressure off the roads. But how will you identify commuters?
    d. Religious building. There is already one place of worship for Catholics in the town. Is there a demand for a second Church (if so, which denomination?), or for a different religious building?
    e. Emergency medical building. Not a full hospital, but a minor injuries centre. This should be built if there are a lot of injuries or future pregnancies likely in the population.
    f. Something else?
2. ***Which one of the following options should be invested in?***
    a. Employment and training. If there is evidence of a lot of unemployment, we should re-train people for new skills.
    b. Old age care. If there is evidence of increasing numbers of retired people in future years, the town will need to allocate more funding for end-of-life care.
    c. Increase spending on schooling. If there is evidence of a growing population of school-aged children (new births, or families moving into the town), then schooling spending should increase.
    d. General infrastructure. If the town is expanding, then services (waste collection; road maintenance, etc.) will require more investment.

Workflow:
In order to answer these questions, the workflow adopted goes thus:

- **Data Inspection:** Data is inspected to check for the current quality of the data. Data Overview, missing values, outliers, blank entries, NaN values, etc are all assessed. This data assessment is done to know what kind of cleaning is required for each feature in the data. The quality of the data is checked based on its validity, uniformity, completeness, accuracy, and consistency as defined on Wikipedia.
- **Data Cleaning:** After inspection, the data is cleaned. Each feature is cleaned and free of inconsistencies that may get in the way of the quality of the data and the analysis of the data.
- **Data Analysis:** Here, Our cleaned data is analysed in order to answer the problem statements.
- **Drawing Insights to make decisions:** Insights gained from the analysis are used to make useful recommendations that answer the problem statements.

INSPECTING THE DATA SET

The census data was loaded and inspected. The overview of the data is as follows:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9655 entries, 0 to 9654
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Unnamed: 0                     9655 non-null   int64
 1   House Number                   9655 non-null   object
 2   Street                         9655 non-null   object
 3   First Name                     9655 non-null   object
 4   Surname                        9655 non-null   object
 5   Age                            9655 non-null   float64
 6   Relationship to Head of House  9655 non-null   object
 7   Marital Status                 7258 non-null   object
 8   Gender                         9655 non-null   object
 9   Occupation                     9655 non-null   object
 10  Infirmity                      9655 non-null   object
 11  Religion                       7189 non-null   object
dtypes: float64(1), int64(1), object(10)
memory usage: 905.3+ KB
```

The data has 12 columns with 9655 entries. Two features have large NaN values. One column *'Unnamed: 0'* is irrelevant to the data. Each of the features is then inspected for inconsistency, invalidity, inaccuracy, non-uniformity and incompleteness.

The '**House Number**' column was characterised with 'string' data type instead of Integer. '**Street**' was correctly populated as there were no missing values, NaN values or 'None' values. '**First Name**' had blank values as well as the '**Last Name'** column. The '**Age**' column, however, was characterised by float values data type. Another inspection worthy of note was the '**Marital Status**' column, where the values were inconsistent. There were abbreviations and NaN values. This phenomenon was also seen in the '**Gender**' column. All other 5 columns were inspected and missing values, blank entries, and None values were observed.

Marital Status:

```
In [1197]: df['Marital Status'].unique()
```

```
Out[1197]: array(['Single', 'Divorced', 'Married', nan, 'Widowed', 'W', 'M', 'D',
                  'S'], dtype=object)
```

```
df['House Number'].unique()
```

```
array(['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12',
       '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23',
       '24', '25', '26', '27', '28', '29', '30', '31', '32', '33', '34',
       '35', '36', '37', '38', '39', '40', '41', '42', '43', '44', '45',
       '46', '47', '48', '49', '50', '51', '52', '53', '54', '55', '56',
       '57', '58', '59', '60', '61', '62', '63', '64', '65', '66', '67',
       '68', '69', '70', '71', '72', '73', '74', '75', '76', '77', '78',
       '79', '80', '81', '82', '83', '84', '85', '86', '87', '88', '89',
       '90', '91', '92', '93', '94', '95', '96', '97', '98', '99', '100',
       '101', '102', '103', '104', '105', '106', '107', '108', '109',
       '110', '111', '112', '113', '114', '115', '116', '117', '118',
       '119', '120', '121', '122', '123', '124', '125', '126', '127',
       '128', '129', '130', '131', '132', '133', '134', '135', '136',
       '137', '138', '139', '140', '141', '142', '143', '144', '145',
       '146', '147', '148', '149', '150', '151', '152', '153', '154',
       '155', '156', '157', '158', '159', '160', '161', '162', '163',
       '164', '165', '166', '167', '168', '169', '170', '171', '172',
       'eight'], dtype=object)
```

```
In [1179]: df['Age'].unique()
```

```
Out[1179]: array([ 56.        ,  43.        ,  30.        ,  53.        ,
                   63.        ,  52.        ,  64.        ,  57.        ,
                   24.        ,  20.        ,  13.        ,  11.        ,
                    9.        ,  45.        ,  60.        ,  55.        ,
                   31.        ,  49.        ,  50.        ,  19.        ,
                   21.        ,  26.        ,  25.        ,  59.        ,
                   46.        ,  41.        ,  14.        ,  16.        ,
                   47.        ,  22.        ,  15.        ,  72.        ,
                   68.        ,  66.        ,  33.        ,  35.        ,
                    5.        ,   2.        ,  86.        ,  37.        ,
                   18.        ,  32.        ,  51.        ,  71.        ,
                   74.        ,  29.        ,   4.        ,  77.        ,
                   83.        ,  65.        ,  79.        ,  40.        ,
                   58.        ,  39.        ,  36.        ,  34.        ,
                   73.        ,  38.        ,  27.        ,  12.        ,
                    7.        ,   8.        ,  81.        ,  80.        ,
                   61.        ,  67.        ,  23.        ,  42.        ,
                   10.        ,  76.        ,  84.        ,  28.        ,
                   17.        ,  69.        ,  62.        ,   6.        ,
                    0.        ,  48.        ,  54.        ,  44.        ,
                    3.        ,  72.04654485,  73.04654485,   1.        ,
                   75.        ,  96.        ,  78.        ,  94.        ,
                   91.        ,  98.        , 102.        ,  70.        ,
                   85.        ,  89.        ,  87.        ,  82.        ,
                   95.        , 103.        ,  97.        ,  88.        ,
                  100.        ,  78.14375872,  90.        ,  92.        ,
                   66.38656872,  68.38656872,  58.62728711,  62.62728711,
                   69.11349969,  68.11349969,  99.        ,  73.95582723,
                   71.95582723,  66.33700147])
```

Gender:

```
In [1206]: df['Gender'].unique()
```

```
Out[1206]: array(['Male', 'Female', 'male', 'M', 'f', 'female', 'm', 'F'],
                 dtype=object)
```

DATA CLEANING

After inspection, it was clear what parts of the data were to be cleaned. The value errors were cleaned thoroughly. A comprehensive review of the cleaning can be viewed in the Jupyter Notebook attached to this report. However, I shall give a summary of the features and how it was cleaned.

1. **Unnamed: 0**: This feature contained values which were the same as the indices of the data set. Hence it was dropped as it was irrelevant to the data.

2. **House Number**: The feature had string values. The numbers were string and it also had an entry written in words - 'eight' instead of '8'. This was cleaned by replacing the words with figures and changing the data type of the values to integer.

3. **Street**: There were no invalidity or inconsistencies observed here.

4. **First Name**: Empty values were cleaned out by checking the 'Last Name' column, 'Age' and 'Relationship to the Head of House'.

5. **Last Name**: The empty value observed here was cleaned by checking out data of people who lived in the same address and the 'Relationship to the Head of House' column to see if there was a family living in the same household and could share the same Last Name.

6. **Age**: All entries here were float type. Age cannot be float. So the data type was changed to integer and that cleaned the column up.

7. **Relationship to Head of House**: Missing values, None values were all cleaned out by checking out other features' values. For example, A person aged less than 18 cannot be the Head of a house. A house/family cannot have two heads too. The blank values were each checked against the Last Name and address to see possible households. For values with same Last Name and address, the missing values was filled with corresponding 'Relationship to Head of House'. For data entries with 'Relationship to Head of House' as None and ages less than 18, Last names and addresses were checked. This gave the direction on how best to clean the data.

8. **Marital Status**: The values entered here were inconsistent. 'M' was replaced by 'Married', 'S' with 'Single', 'D' with 'Divorced', and 'W' with 'Widowed'. NaN here were cleaned out by checking for the Age of the people in this category. There

were all less than 18 and could be categorised as 'Minor' as the legal marriageable age in the UK. (GOV.UK, 2022)
Furthermore, It was observed that three entries had Marital Status as 'Divorced' despite being 16 years of age. Checking the 'Last Name' reveals that these data all have children that they live with in the same address. This is illegal and very inconsistent. Other columns in these data like 'Relationship to Head of House' were checked for further correlation and dropping the data was the best way to clean the data. Three households with 6 entries in total were dropped as they are insignificant to our overall analysis.

9. **Gender**: There were inconsistent values recorded here too. The data entries were cleaned appropriately with accurate data values based on the existing values. 'Male' and 'Female' were the accurate values expected.

10. **Occupation**: Blank values were observed and replaced as 'Unknown'. There was individual occupation. However individual data records like 'Child Physiotherapist', and 'Meteorologist' won't aid in our analysis. So it is better to create a new column that showcases the 'Employment Status' of the data entry.

11. **Employment Status**: This was created by filtering out data from the 'Occupation' column. It consists of 'Employed', 'Unemployed', 'Student', 'University Student', 'Child' and 'Retired'.

12. **Infirmity**: Blank values and 'None' values were observed. 'None' was cleaned out as 'No disability', and 'Blank values', upon checking other columns and how it correlates and replaced with 'No Disability'.

13. **Religion**: People with Ages more than 18 and 'None' were classified as 'Irreligious' as at that period, they ought to have made such a decision on what religion they wanted to follow. People less than 18 however were categorised as 'NA' as they are too young to form decisions on what religion to practice.

DATA ANALYSIS

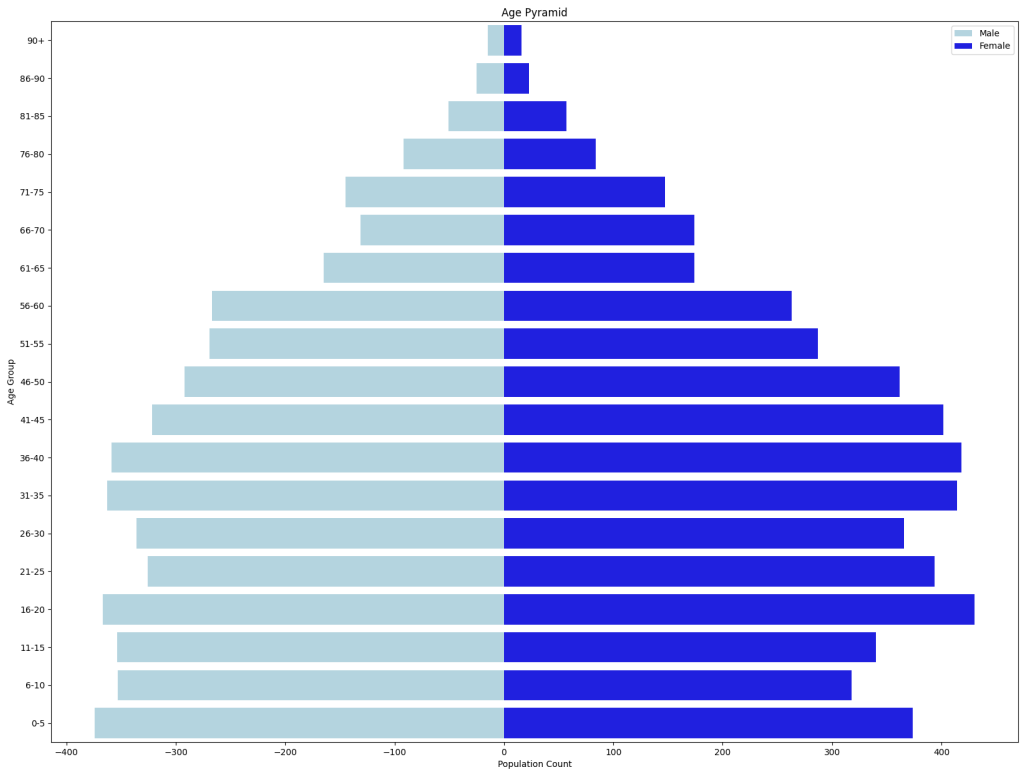
The data now look like this after cleaning:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9649 entries, 0 to 9648
Data columns (total 16 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   House Number                  9649 non-null   int32
 1   Street                        9649 non-null   object
 2   First Name                    9649 non-null   object
 3   Last Name                     9649 non-null   object
 4   Age                           9649 non-null   int32
 5   Relationship to Head of House 9649 non-null   object
 6   Marital Status                9649 non-null   object
 7   Gender                        9649 non-null   object
 8   Occupation                    9649 non-null   object
 9   Infirmity                     9649 non-null   object
 10  Religion                      9649 non-null   object
 11  Employment Status             9649 non-null   object
 12  Age Group                     9649 non-null   object
 13  Address                       9649 non-null   object
 14  No of Occupants               9649 non-null   int64
 15  Occupancy Level               9649 non-null   object
dtypes: int32(2), int64(1), object(13)
memory usage: 1.2+ MB
```

New columns were added to assist in our analysis. They are:

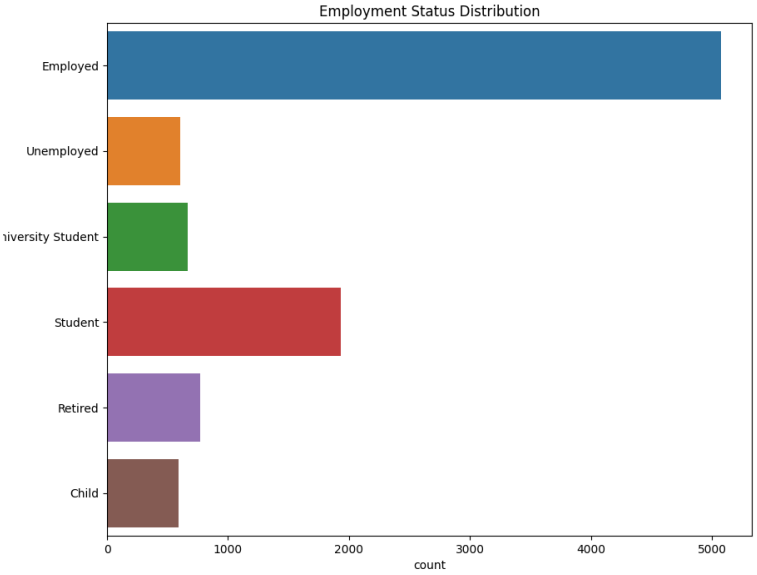
- **Age Group:** 5-year groups for the population pyramid
- **Employment Status:** With 'Employed', 'Unemployed', Student, 'University Student', and 'Retired' as values.
- **No of Occupants:** Number of Occupants per household.
- **Occupancy Level**

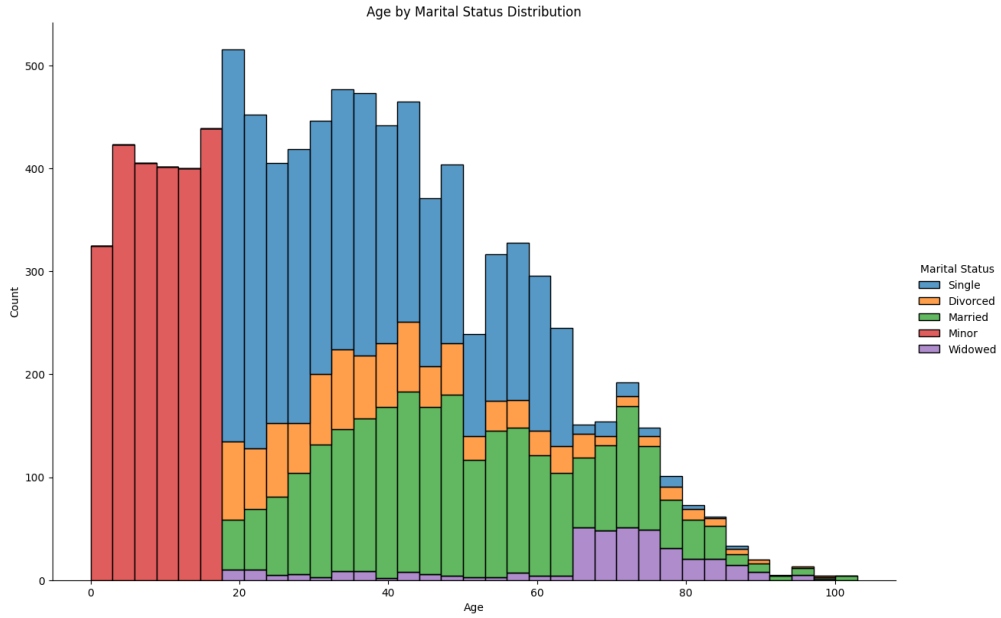
The population pyramid shows that there's a high number of young people and more

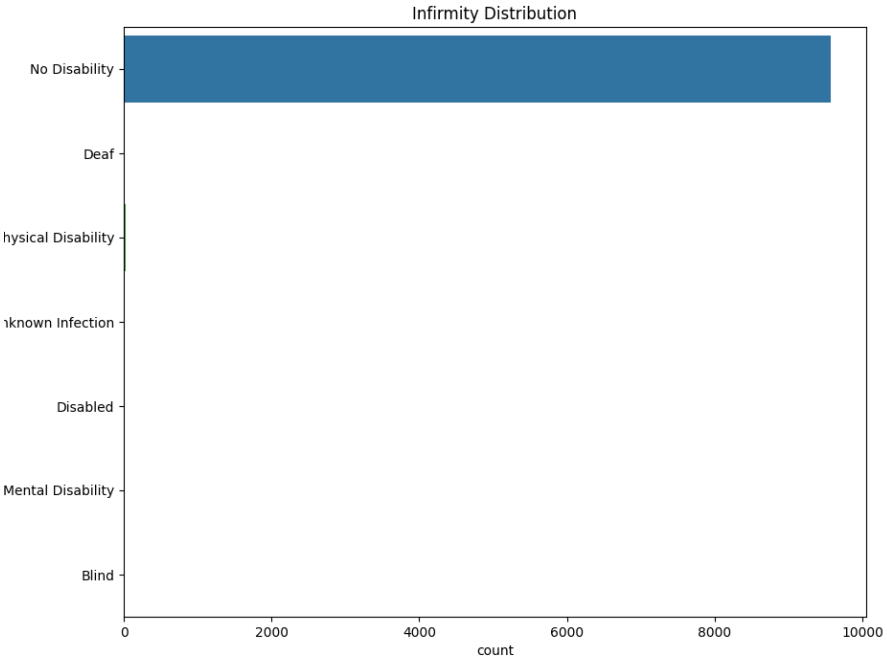


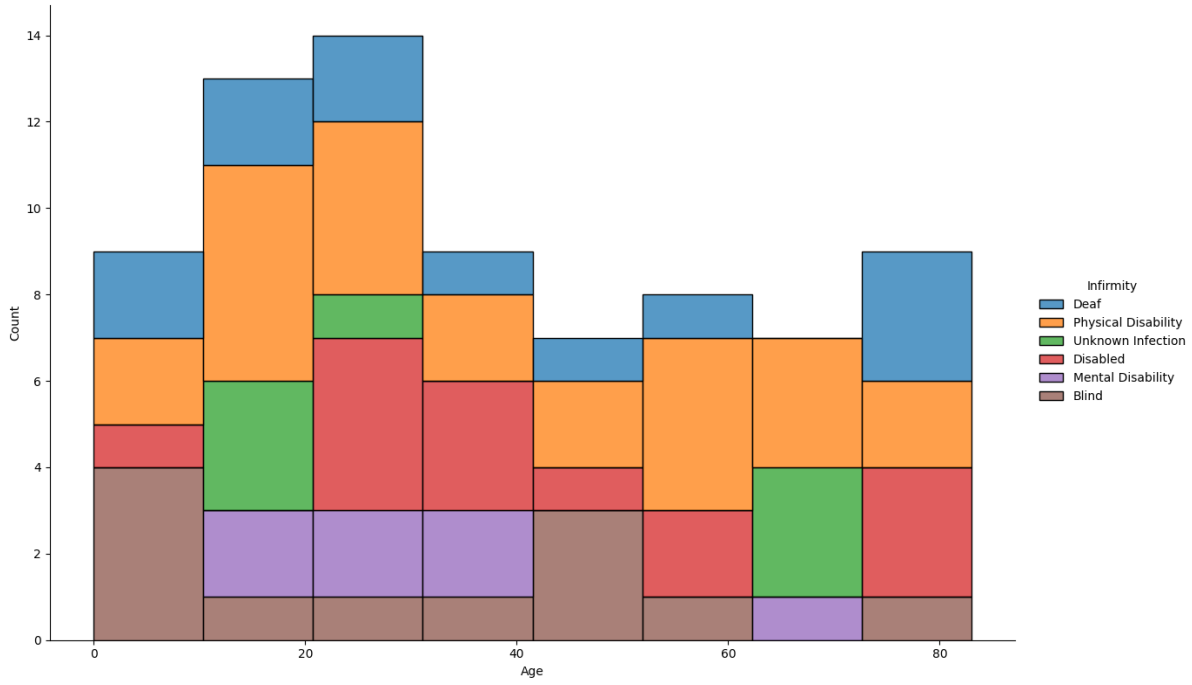
females between ages 16 -20.

The analysis goes on to show us that about 52% of the population is Employed, Students(Including university students) are about 26% of the population.Only 6.3% are unemployed. Being a young population, this town is characterised with highest number of people single (35%) with only about 9% Divorced spread across all ages from 20. About 99% of the total population have No disability at all and 'Physical Disability' occurs accross all ages.


Employment Status Distribution
Employed
Unemployed
iversity Student
Student
Retired
Child
0
1000
2000
3000
4000
5000
count

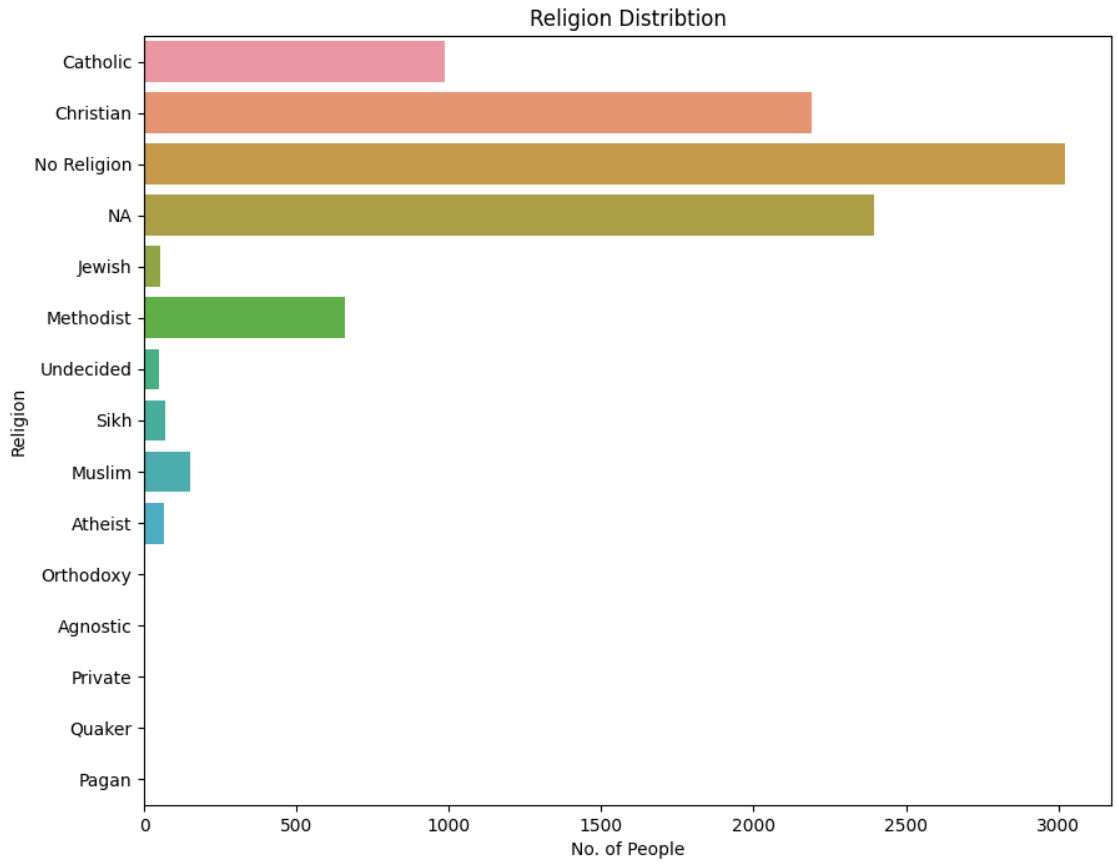

Age by Marital Status Distribution
Count
Age
0
20
40
60
80
100
0
100
200
300
400
500
Marital Status
Single
Divorced
Married
Minor
Widowed


Infirmity Distribution
No Disability
Deaf
hysical Disability
nknown Infection
Disabled
Mental Disability
Blind
0
2000
4000
6000
8000
10000
count


Count
Age
0
20
40
60
80
0
2
4
6
8
10
12
14
Infirmity
Deaf
Physical Disability
Unknown Infection
Disabled
Mental Disability
Blind

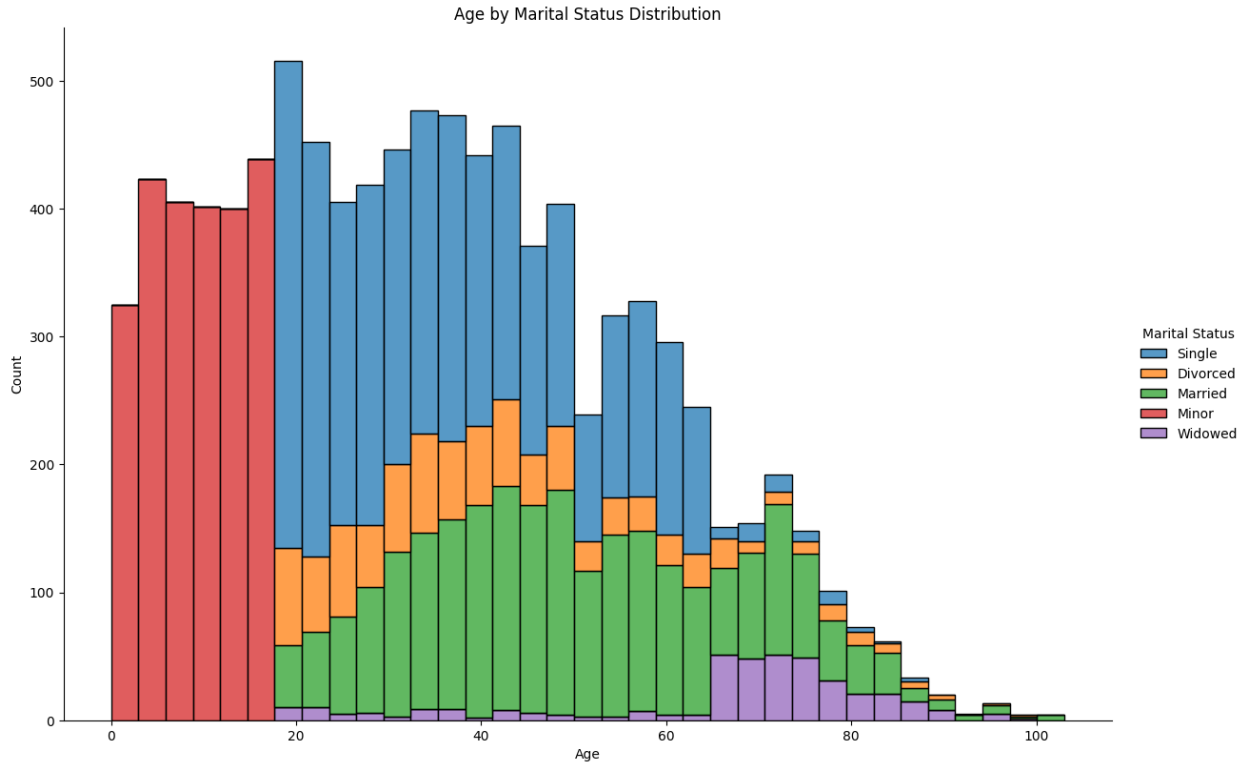
Answering the Problem Questions:

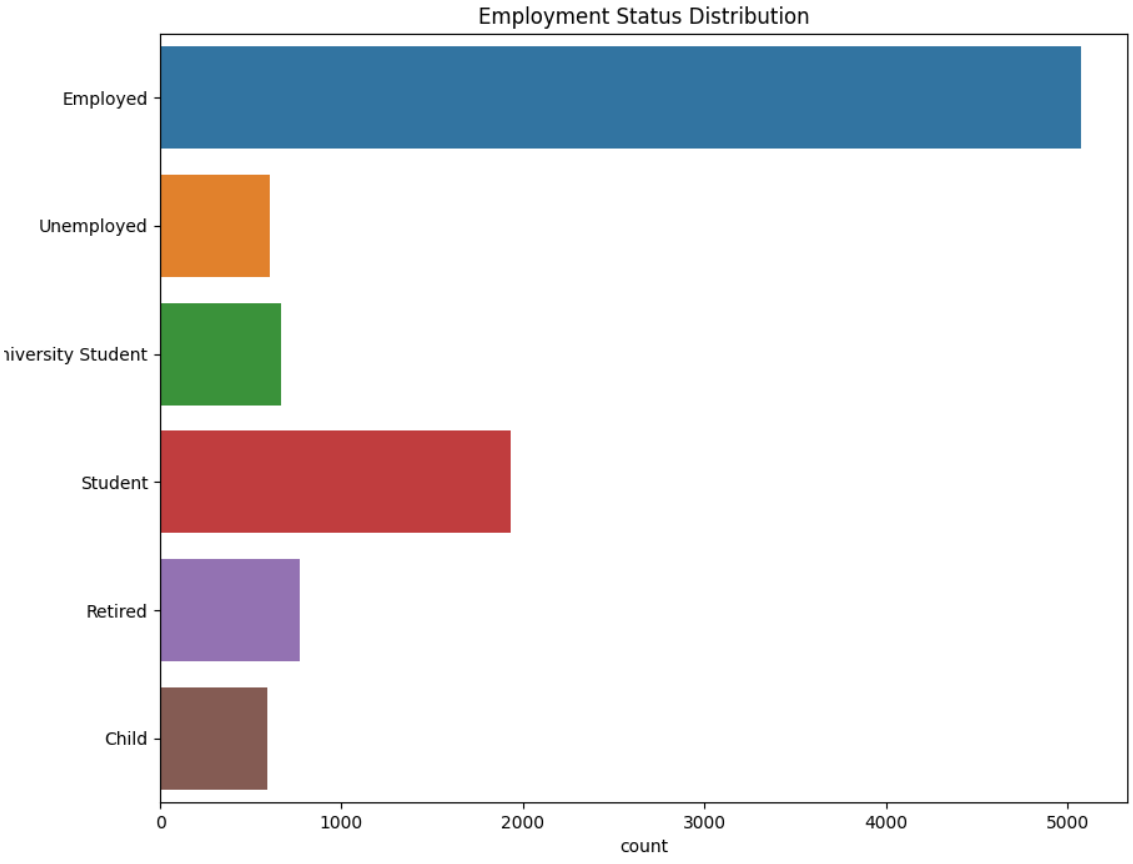
A new emergency medical building would not be a good investment as 99% of the people who live in the town have no medical emergencies to cater for. For religions, having them commute out of town to their places of worship works as the largest of the population do not identify with any religion.



For the population with a large number of single people between ages 20 and 40, it is projected that a population increase might be due. Therefore, a high density housing might a yielding investment for the government to look into.

However, because a larger part of the population are employed, followed by students, a Train Station should be built on an unoccupied plot of land that the local government wishes to develop.


Age by Marital Status Distribution
Count
Age
0
100
200
300
400
500
0
20
40
60
80
100
Marital Status
Single
Divorced
Married
Minor
Widowed


Employment Status Distribution
Employed
Unemployed
iversity Student
Student
Retired
Child
0
1000
2000
3000
4000
5000
count

6.3% of the population, all with ages greater than 18 are unemployed. This is significantly higher than the average unemployment rate in the UK which is 4.53% (MacroTrends, 2022). Hence, a worthy investment is skills and empowerment trainings for the unemployed in the town.

General Infrastructure to improve social life in the town is also a worthy investment. Seeing as there is a larger percentage of Singles older than 20, a public place or event might be necessary to improve the social life and help inhabitants find happiness by finding partners. (Better Health, 2022)

References

1. *Data Cleansing, Data Quality, Wikipedia (https://en.wikipedia.org/wiki/Data_cleansing#Data_quality)*
2. *United Kingdom Unemployment Rate, published in 2022 on (https://www.macrotrends.net/countries/GBR/united-kingdom/unemployment-rate)*
3. *Implementation of the Marriage and Civil Partnership(Minimum Age) Act 2022 (https://www.gov.uk/government/news/implementation-of-the-marriage-and-civil-partnership-minimum-age-act-2022#:~:text=The%20Marriage%20and%20Civil%20Partnership%20(Minimum%20Age)%20Act%202022%20received,on%20Monday%2027%20February%202023.&text=The%20Act%20will%20raise%20the,the%20scourge%20of%20forced%20marriage)*
4. *Strong Relationships, Strong Health,2022 (https://www.betterhealth.vic.gov.au/health/healthyliving/Strong-relationships-strong-health)*