Dhaka City People information Analysis

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df=pd.read_csv("Dhaka_people.csv")
df.head()
```

Out[2]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	Female	45.0	0.0	0.0	0.0	Mirpur	Undeveloped	Building	Dhaka	0.0
1	Male	17.0	0.0	0.0	1.0	Chawkbazar	Developed	Building	Dhaka	0.0
2	Female	29.0	0.0	0.0	0.0	Paltan	Undeveloped	Other	Dhaka	0.0
3	Female	63.0	1.0	1.0	0.0	Motijheel	Developed	Other	Dhaka	1.0
4	Male	22.0	0.0	0.0	0.0	Gendaria	Undeveloped	Building	Dhaka	0.0

Label_Encoder

In [3]:

```
from sklearn.preprocessing import LabelEncoder
led =LabelEncoder()
led.fit_transform(df['Gender'])
df['Gender'] =led.fit_transform(df['Gender'])
df.head()
```

Out[3]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	0	45.0	0.0	0.0	0.0	Mirpur	Undeveloped	Building	Dhaka	0.0
1	1	17.0	0.0	0.0	1.0	Chawkbazar	Developed	Building	Dhaka	0.0
2	0	29.0	0.0	0.0	0.0	Paltan	Undeveloped	Other	Dhaka	0.0
3	0	63.0	1.0	1.0	0.0	Motijheel	Developed	Other	Dhaka	1.0
4	1	22.0	0.0	0.0	0.0	Gendaria	Undeveloped	Building	Dhaka	0.0

In [4]:

```
df.shape
```

Out[4]:
```
(1001, 10)
```

In [5]:
```
df.isnull()
```

Out[5]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
996	False	False	False	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False	False	False	False
1000	False	True	True	True	True	True	True	True	True	True

1001 rows × 10 columns

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Gender       0
Age          1
NS1          1
IgG          1
IgM          1
Area         1
AreaType     1
HouseType    1
District     1
Outcome      1
dtype: int64
```

In [7]:
```
from sklearn.model_selection import train_test_split
train , test = train_test_split(df,test_size=.70, random_state=42)
```

In [8]:
```
train.shape
```

Out[8]:
```
(300, 10)
```

In [9]:

```
test.shape
```

Out[9]:

```
(701, 10)
```

In [10]:

```
test.head()
```

Out[10]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
521	1	23.0	1.0	1.0	0.0	Kamrangirchar	Developed	Tinshed	Dhaka	1.0
941	0	37.0	0.0	0.0	0.0	Rampura	Developed	Building	Dhaka	0.0
741	1	65.0	0.0	0.0	0.0	Khilgaon	Developed	Tinshed	Dhaka	0.0
980	1	11.0	0.0	0.0	0.0	Banasree	Undeveloped	Other	Dhaka	0.0
411	1	24.0	0.0	0.0	1.0	Hazaribagh	Developed	Other	Dhaka	0.0

In [11]:

```
test.to_csv('dhaka_testing.csv')
```

In [12]:

```
df1 =df.copy()
df2 =df.copy()
df3 =df.copy()
```

In [13]:

```
df.head()
```

Out[13]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	0	45.0	0.0	0.0	0.0	Mirpur	Undeveloped	Building	Dhaka	0.0
1	1	17.0	0.0	0.0	1.0	Chawkbazar	Developed	Building	Dhaka	0.0
2	0	29.0	0.0	0.0	0.0	Paltan	Undeveloped	Other	Dhaka	0.0
3	0	63.0	1.0	1.0	0.0	Motijheel	Developed	Other	Dhaka	1.0
4	1	22.0	0.0	0.0	0.0	Gendaria	Undeveloped	Building	Dhaka	0.0

In [14]:

```
df['Gender'].value_counts()
```

Out[14]:

```
0    524
1    476
2      1
Name: Gender, dtype: int64
```

In [15]:

```
Female=(524/(524+476))*100
Male =(476/(524+476))*100
print('Female {} percent of total People '.format(Female))
print('Male  {} percent of total People '.format(Male))
```

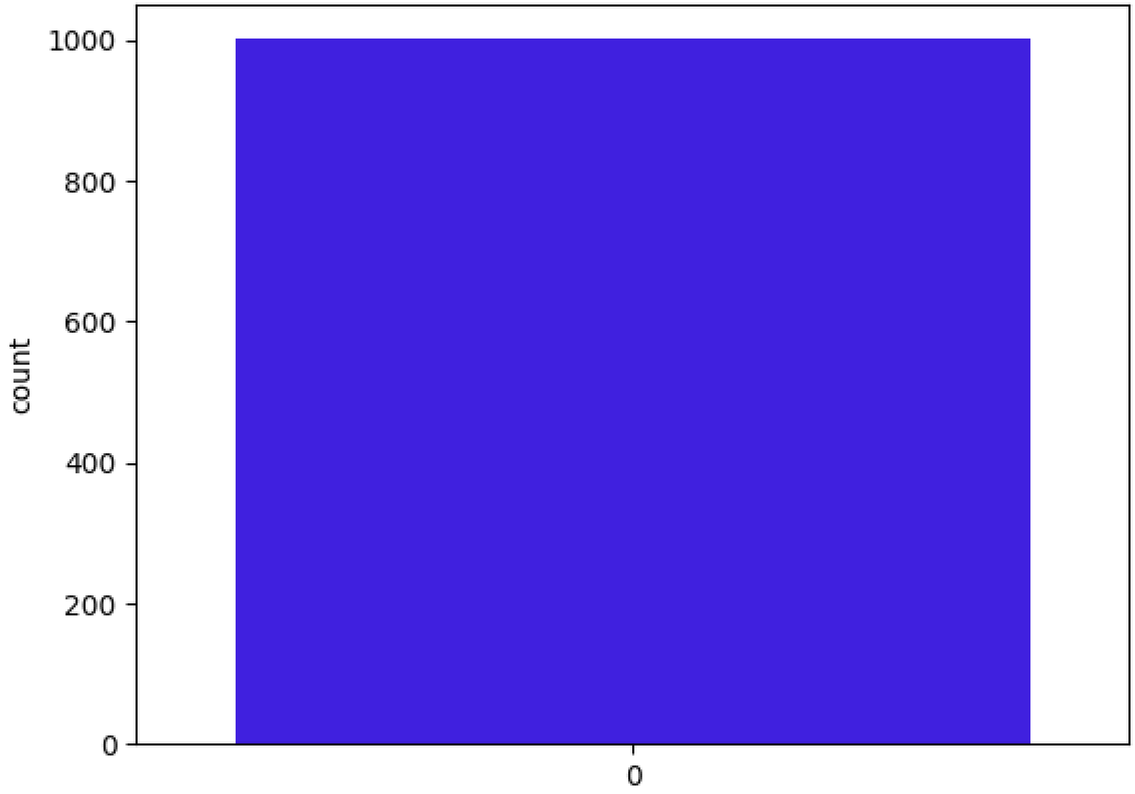
```
Female 52.400000000000006 percent of total People
Male  47.599999999999994 percent of total People
```

In [16]:

```
sns.countplot(df['Gender'],color='#2B00FF')
```

Out[16]:

```
<Axes: ylabel='count'>
```



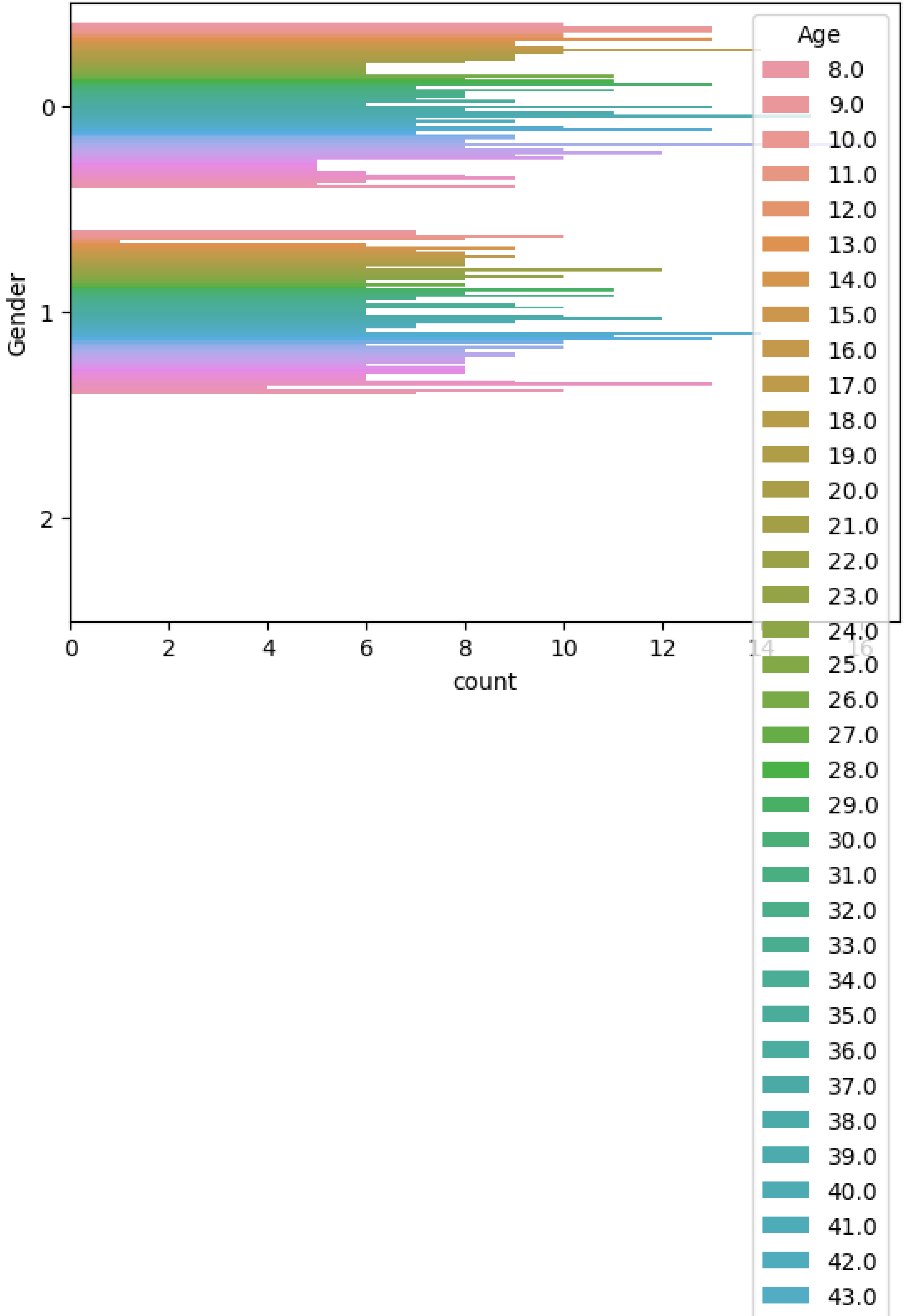
Basic Visualization using Seaborn Library

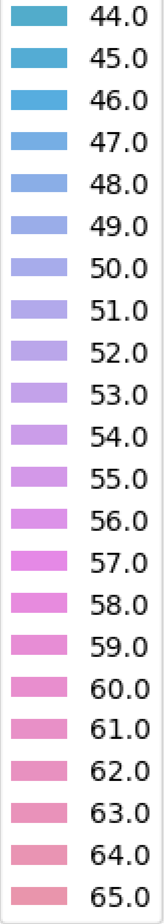
In [17]:

```
sns.countplot( y = 'Gender', hue='Age', data=df )
```

Out[17]:

```
<Axes: xlabel='count', ylabel='Gender'>
```


Age
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
24.0
25.0
26.0
27.0
28.0
29.0
30.0
31.0
32.0
33.0
34.0
35.0
36.0
37.0
38.0
39.0
40.0
41.0
42.0
43.0
Gender
0
1
2
count
0
2
4
6
8
10
12
14
16

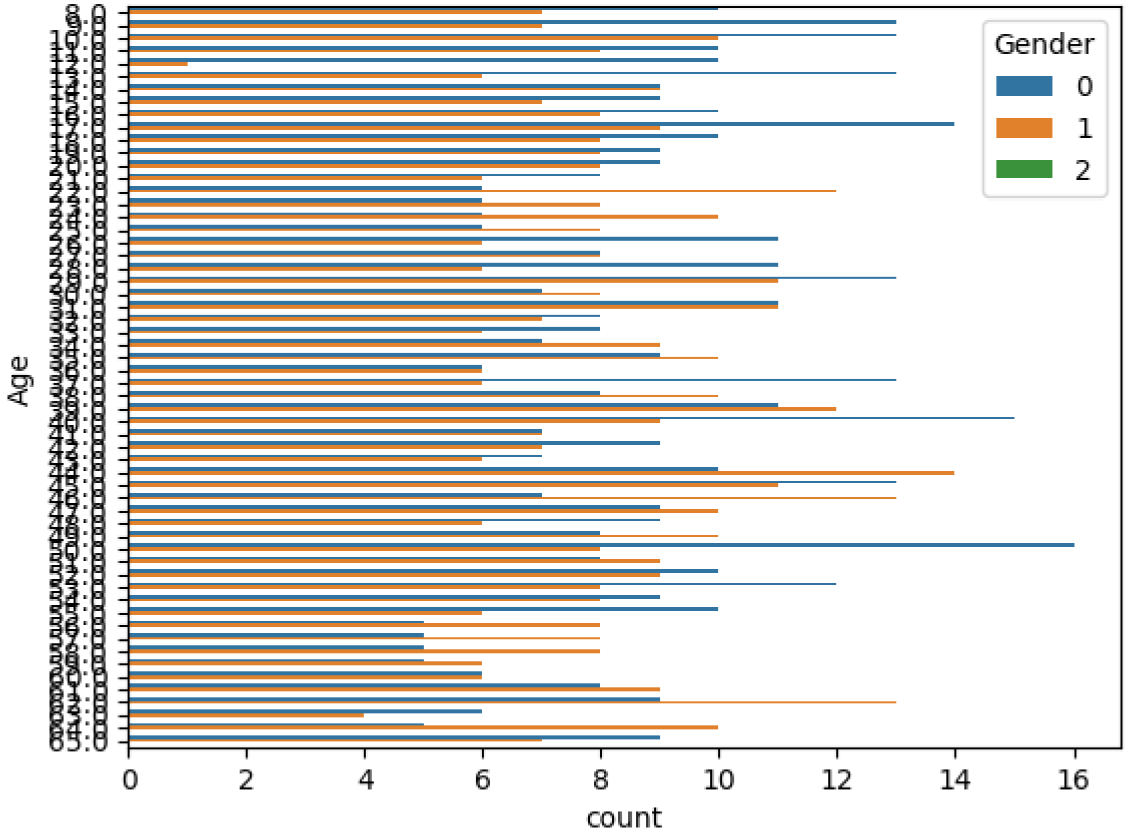


In [45]:

```
sns.countplot( y = 'Age', hue='Gender', data=df )
```

Out[45]:

```
<Axes: xlabel='count', ylabel='Age'>
```

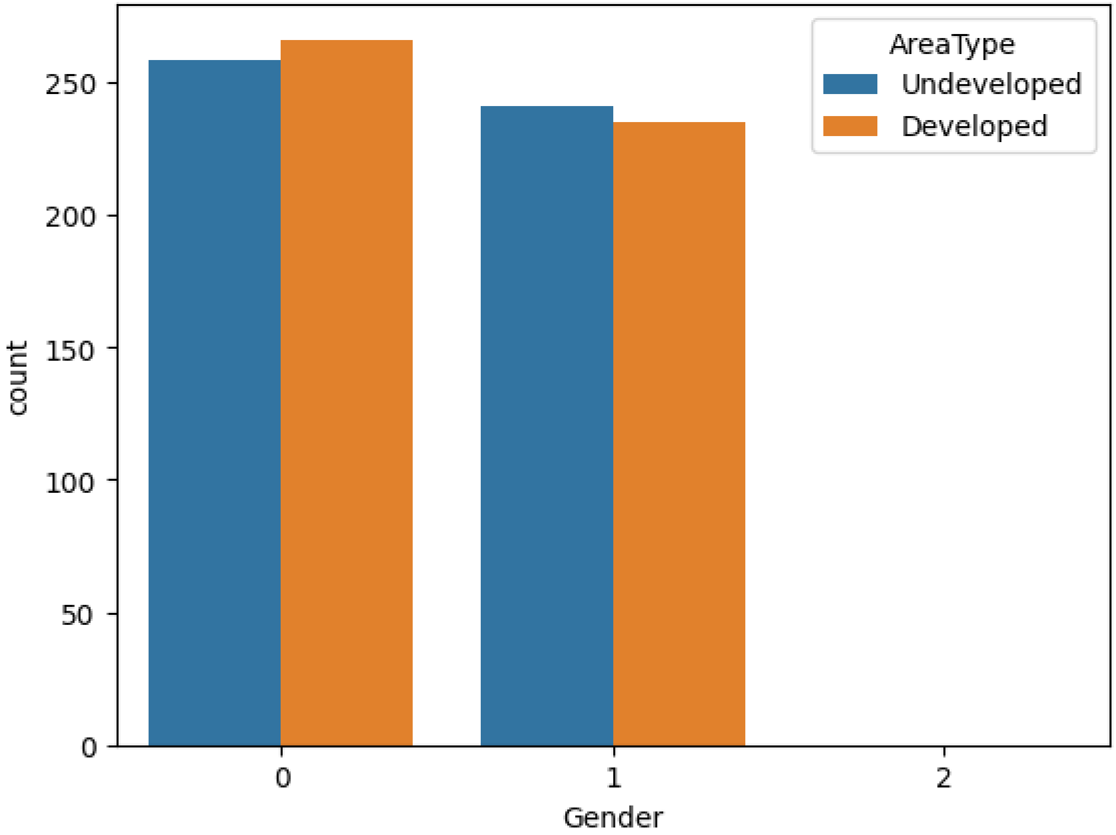


In [19]:

```
sns.countplot( x = 'Gender', hue='AreaType', data=df )
```

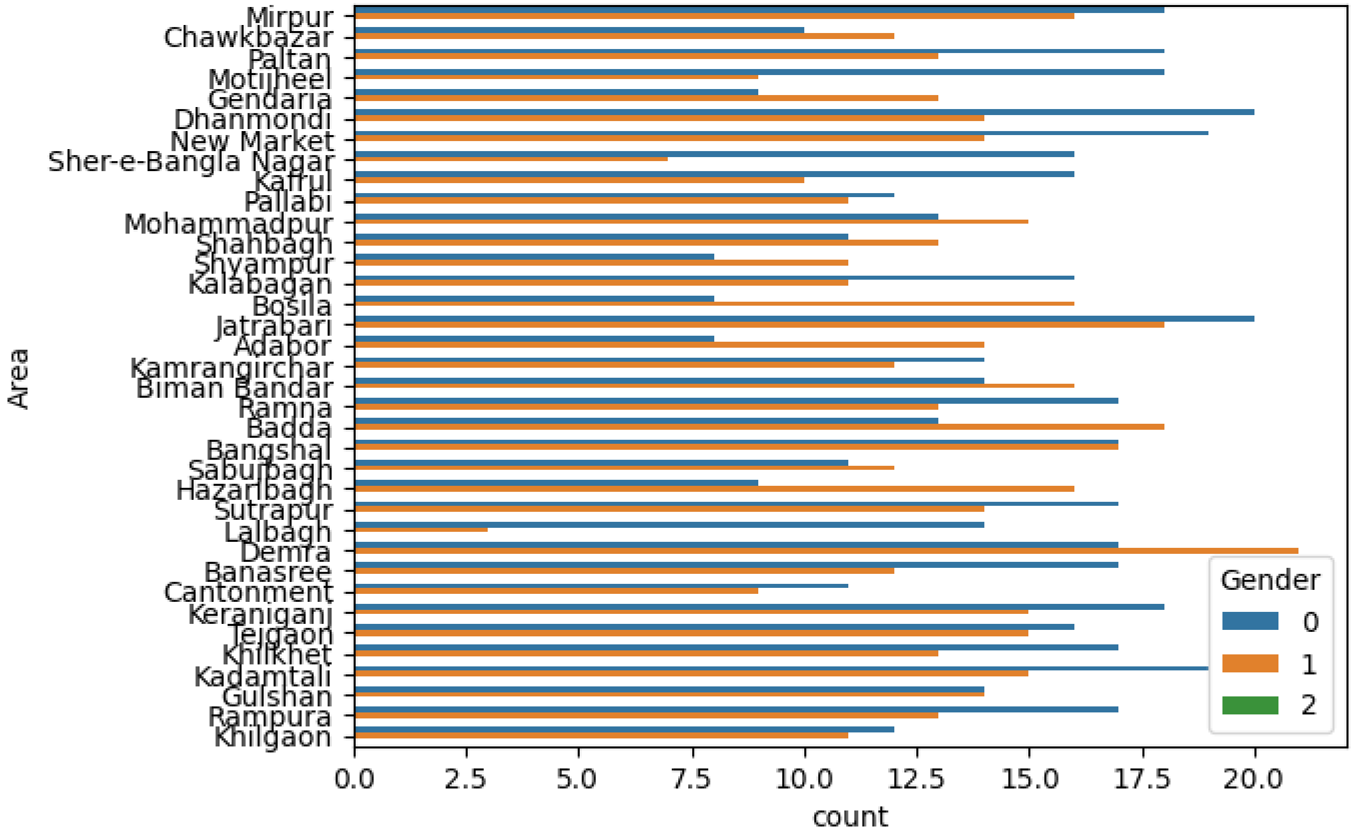
Out[19]:

```
<Axes: xlabel='Gender', ylabel='count'>
```



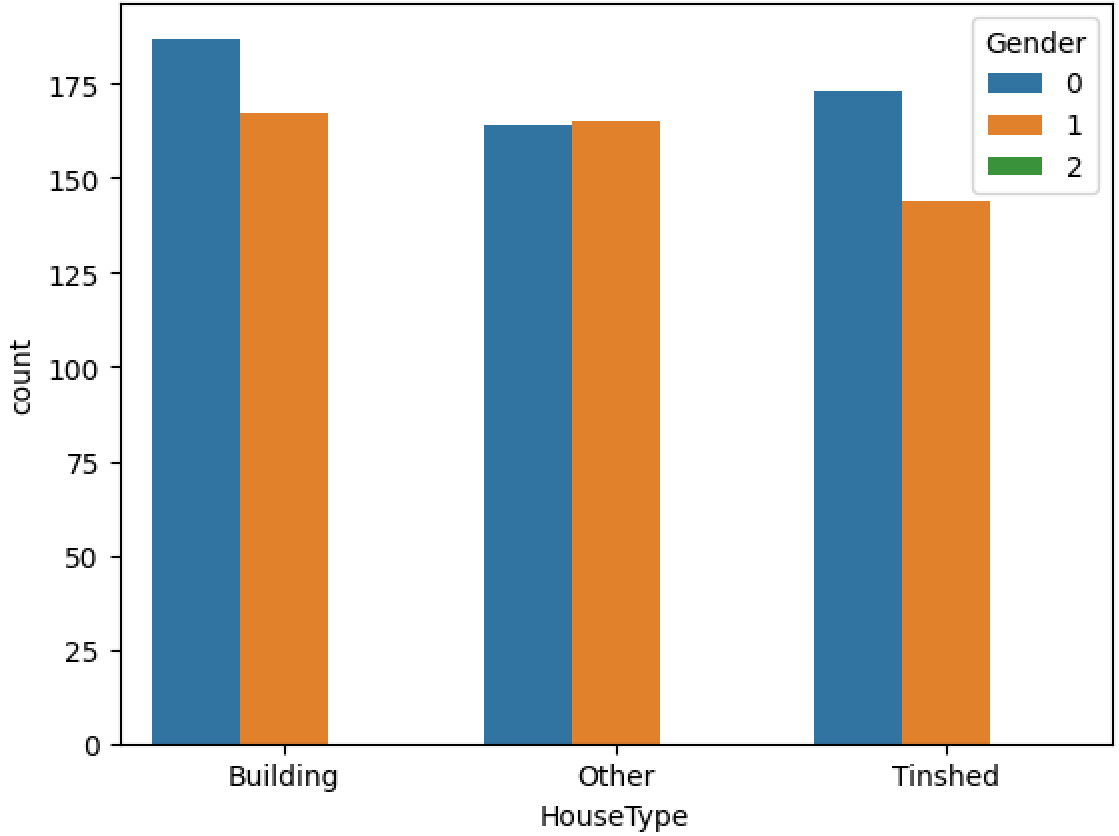
```
In [20]: sns.countplot( y = 'Area', hue='Gender', data=df )
```

```
Out[20]: <Axes: xlabel='count', ylabel='Area'>
```



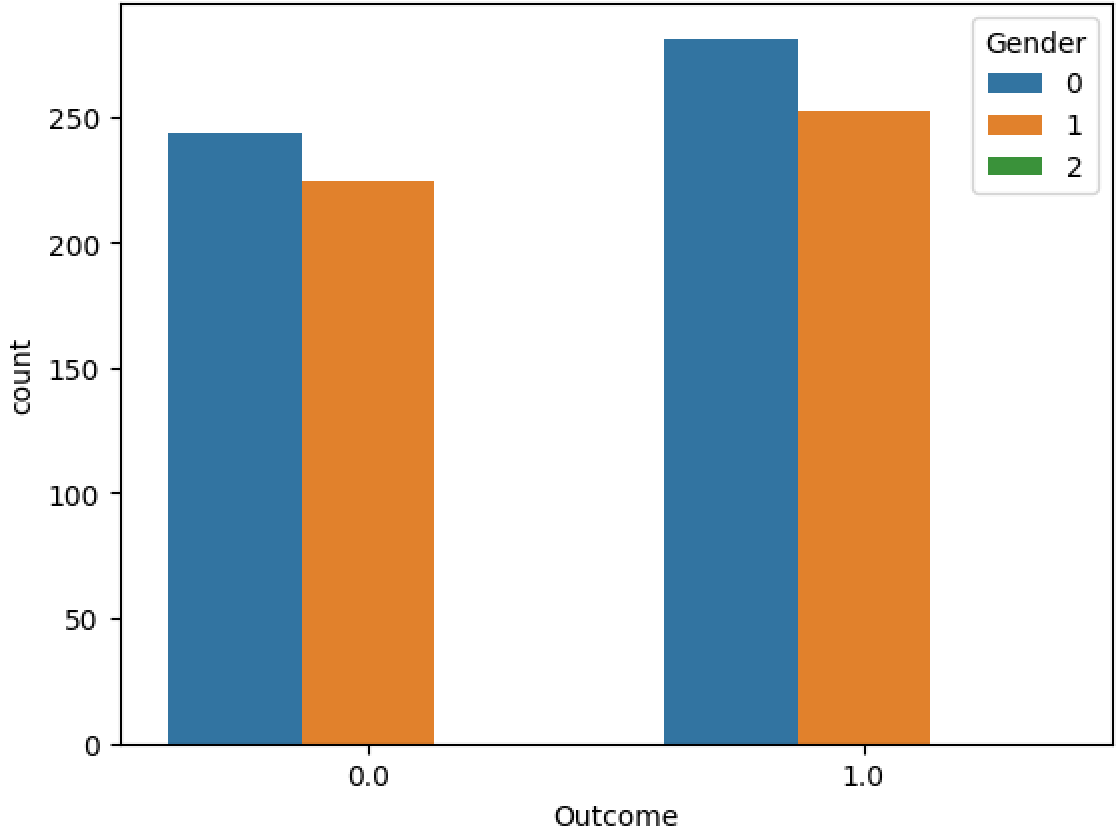
In [21]:
```
sns.countplot( x = 'HouseType', hue='Gender', data=df )
```

Out[21]:
```
<Axes: xlabel='HouseType', ylabel='count'>
```



In [22]:
```
sns.countplot( x = 'Outcome', hue='Gender', data=df )
```

Out[22]:
```
<Axes: xlabel='Outcome', ylabel='count'>
```

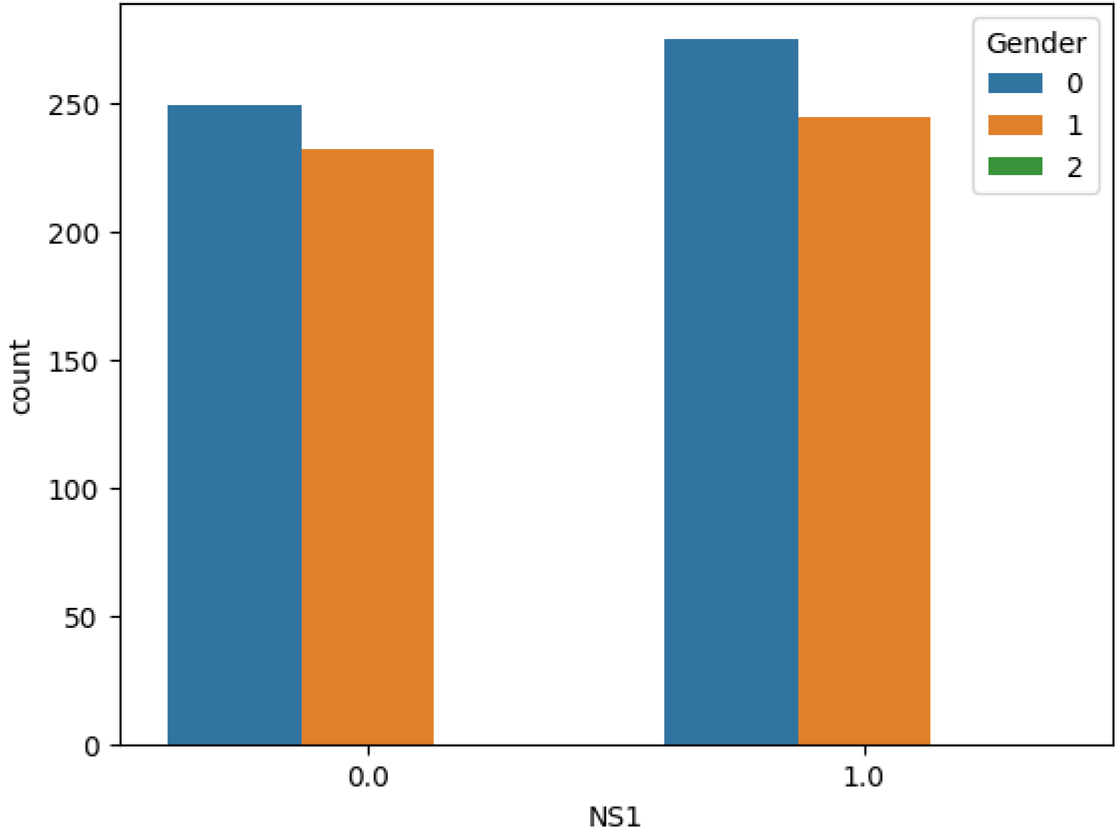


In [23]:

```
sns.countplot( x = 'NS1', hue='Gender', data=df )
```

Out[23]:

```
<Axes: xlabel='NS1', ylabel='count'>
```



In [24]:

```
df['AreaType'].value_counts()
```

Out[24]:

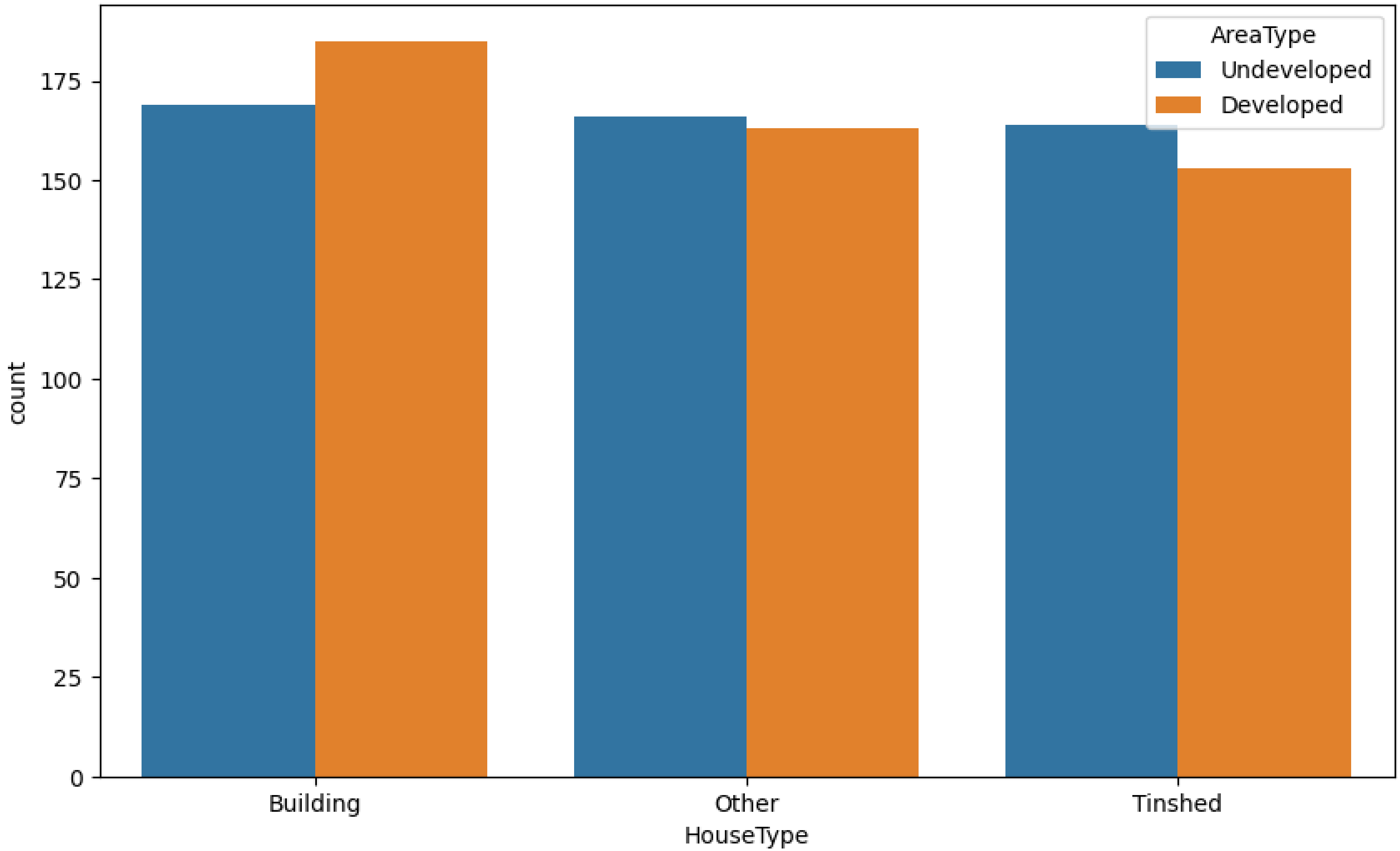
```
Developed      501
Undeveloped    499
Name: AreaType, dtype: int64
```

In [25]:

```
plt.subplots(figsize=(10,6))
sns.countplot(x = 'HouseType', hue='AreaType', data=df)
```

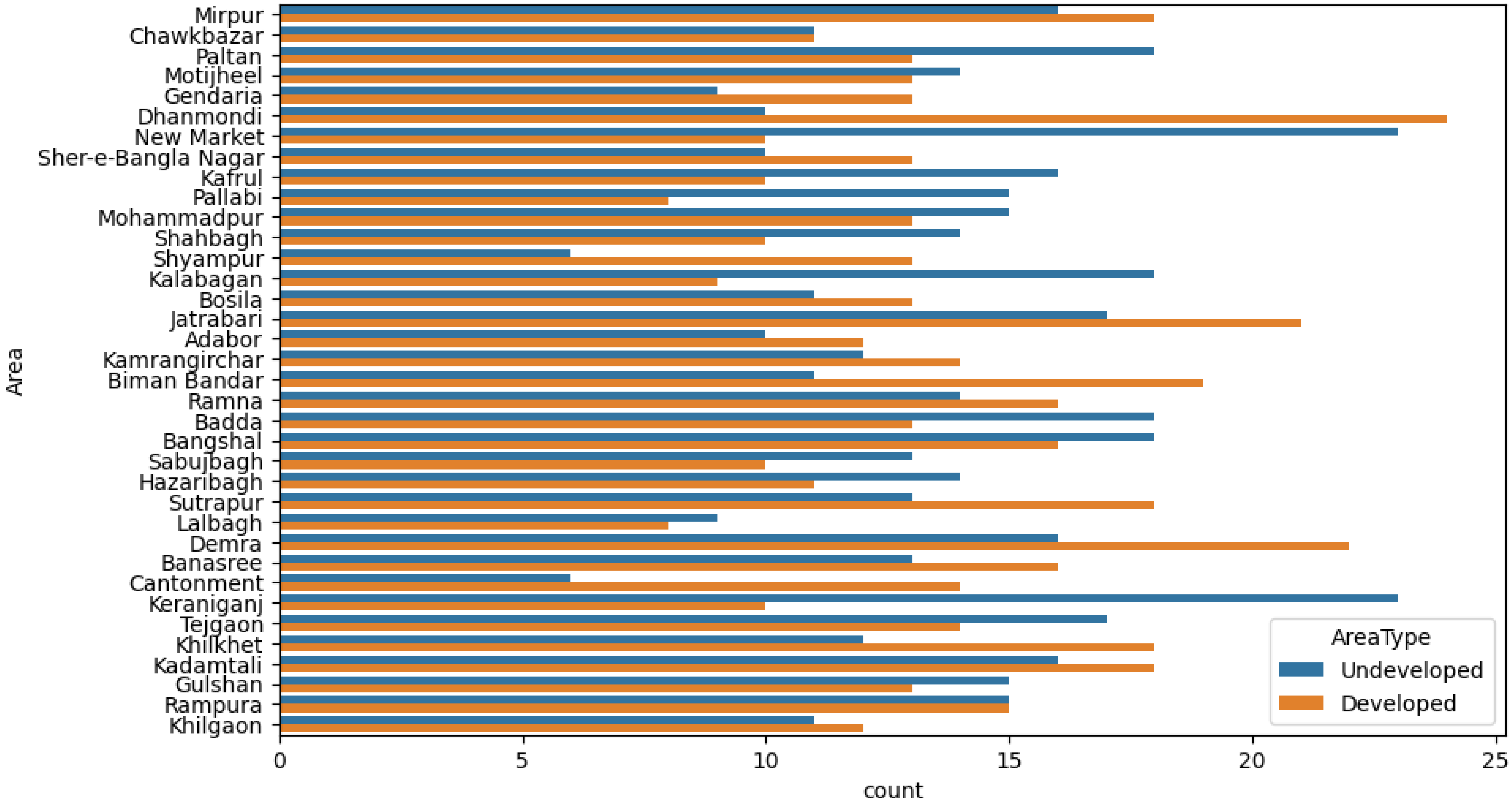
Out[25]:

```
<Axes: xlabel='HouseType', ylabel='count'>
```



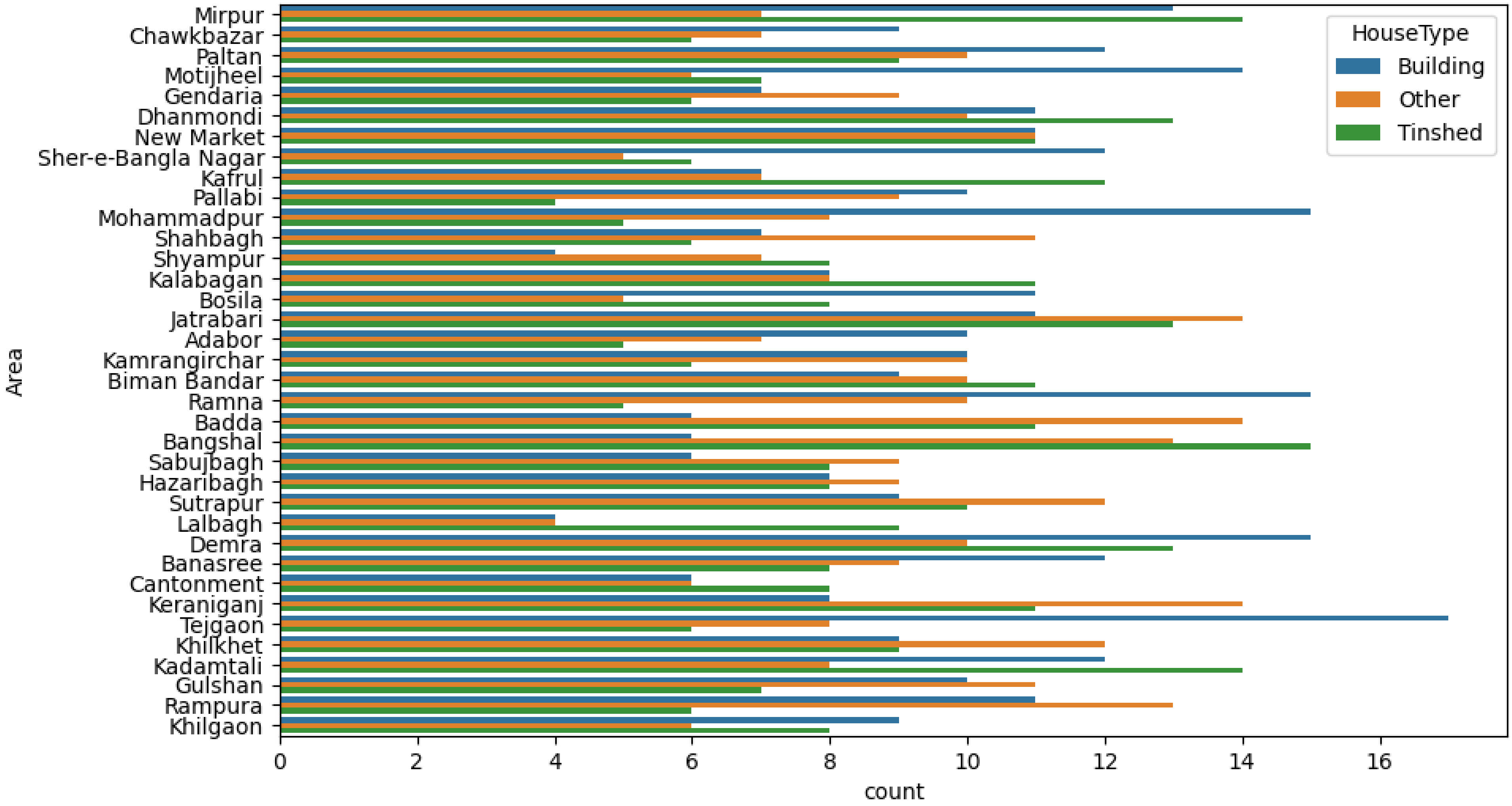
```
In [26]: plt.subplots(figsize=(10,6))
         sns.countplot(y = 'Area', hue='AreaType', data=df)
```

```
Out[26]: <Axes: xlabel='count', ylabel='Area'>
```



```
In [27]: plt.subplots(figsize=(10,6))
         sns.countplot(y= 'Area', hue='HouseType', data=df)
```

```
Out[27]: <Axes: xlabel='count', ylabel='Area'>
```



Encoder of Label_encoder

In [28]:

```
from sklearn.preprocessing import LabelEncoder
label =LabelEncoder()
```

In [29]:

```
df1.columns
```

Out[29]:

```
Index(['Gender', 'Age', 'NS1', 'IgG', 'IgM', 'Area', 'AreaType', 'HouseType',
       'District', 'Outcome'],
      dtype='object')
```

In [30]:

```
from pandas.core.dtypes.common import is_numeric_dtype
for column in df1.columns:
    if is_numeric_dtype(df1[column]):
        continue
    else:
        df1[column] = label.fit_transform(df1[column])
```

In [31]:
```
df1.head()
```

Out[31]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	0	45.0	0.0	0.0	0.0	22	1	0	0	0.0
1	1	17.0	0.0	0.0	1.0	7	0	0	0	0.0
2	0	29.0	0.0	0.0	0.0	27	1	1	0	0.0
3	0	63.0	1.0	1.0	0.0	24	0	1	0	1.0
4	1	22.0	0.0	0.0	0.0	10	1	0	0	0.0

Pandas Profiling & pie

In [32]:
```
import pandas as pd
import numpy as np
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import warnings
warnings.filterwarnings('ignore')
```

In [33]:
```
df.head()
```

Out[33]:

	Gender	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	0	45.0	0.0	0.0	0.0	Mirpur	Undeveloped	Building	Dhaka	0.0
1	1	17.0	0.0	0.0	1.0	Chawkbazar	Developed	Building	Dhaka	0.0
2	0	29.0	0.0	0.0	0.0	Paltan	Undeveloped	Other	Dhaka	0.0
3	0	63.0	1.0	1.0	0.0	Motijheel	Developed	Other	Dhaka	1.0
4	1	22.0	0.0	0.0	0.0	Gendaria	Undeveloped	Building	Dhaka	0.0

In [34]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
```

```
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Gender     1001 non-null   int32  
 1   Age        1000 non-null   float64
 2   NS1        1000 non-null   float64
 3   IgG        1000 non-null   float64
 4   IgM        1000 non-null   float64
 5   Area       1000 non-null   object 
 6   AreaType   1000 non-null   object 
 7   HouseType  1000 non-null   object 
 8   District   1000 non-null   object 
 9   Outcome    1000 non-null   float64
dtypes: float64(5), int32(1), object(4)
memory usage: 74.4+ KB
```

In [35]:

```python
x =df.drop('Gender',axis=1)
y = df[['Gender']]
```

In [36]:

```python
x.head()
```

Out[36]:

	Age	NS1	IgG	IgM	Area	AreaType	HouseType	District	Outcome
0	45.0	0.0	0.0	0.0	Mirpur	Undeveloped	Building	Dhaka	0.0
1	17.0	0.0	0.0	1.0	Chawkbazar	Developed	Building	Dhaka	0.0
2	29.0	0.0	0.0	0.0	Paltan	Undeveloped	Other	Dhaka	0.0
3	63.0	1.0	1.0	0.0	Motijheel	Developed	Other	Dhaka	1.0
4	22.0	0.0	0.0	0.0	Gendaria	Undeveloped	Building	Dhaka	0.0

In [37]:

```python
y.head()
```

Out[37]:

	Gender
0	0
1	1
2	0
3	0
4	1

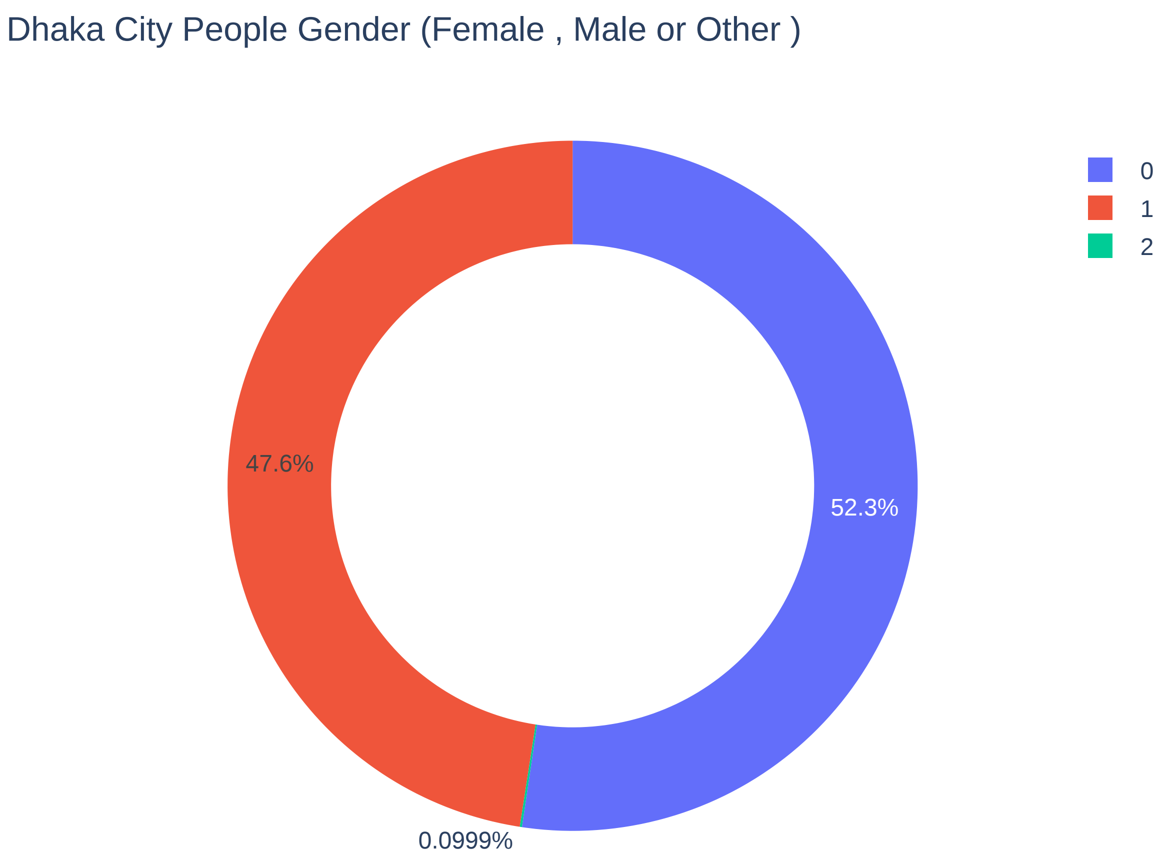
In [38]:

```python
print('Gander  in 100%')
round(df.Gender.value_counts()*100/len(df),1)
```

```
Out[38]: Gander  in 100%
         0    52.3
         1    47.6
         2     0.1
         Name: Gender, dtype: float64
```

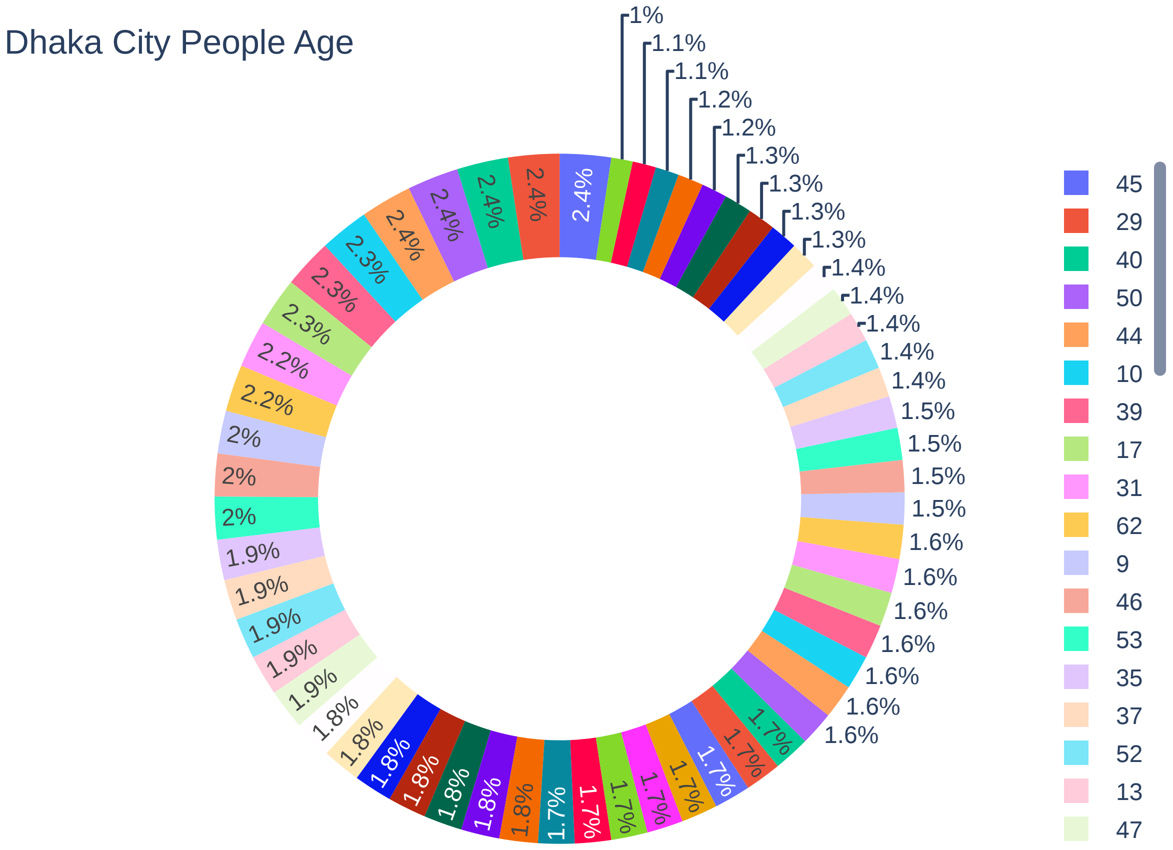
Vaz

```
In [39]: Gender =df['Gender'].value_counts()
         transctions = Gender.index
         quantity =Gender.values
         figure =px.pie(df,
                       values=quantity,
                       names =transctions,hole=.70,
                       title=" Dhaka City People Gender (Female , Male or Other ) ")

         figure.show()
```



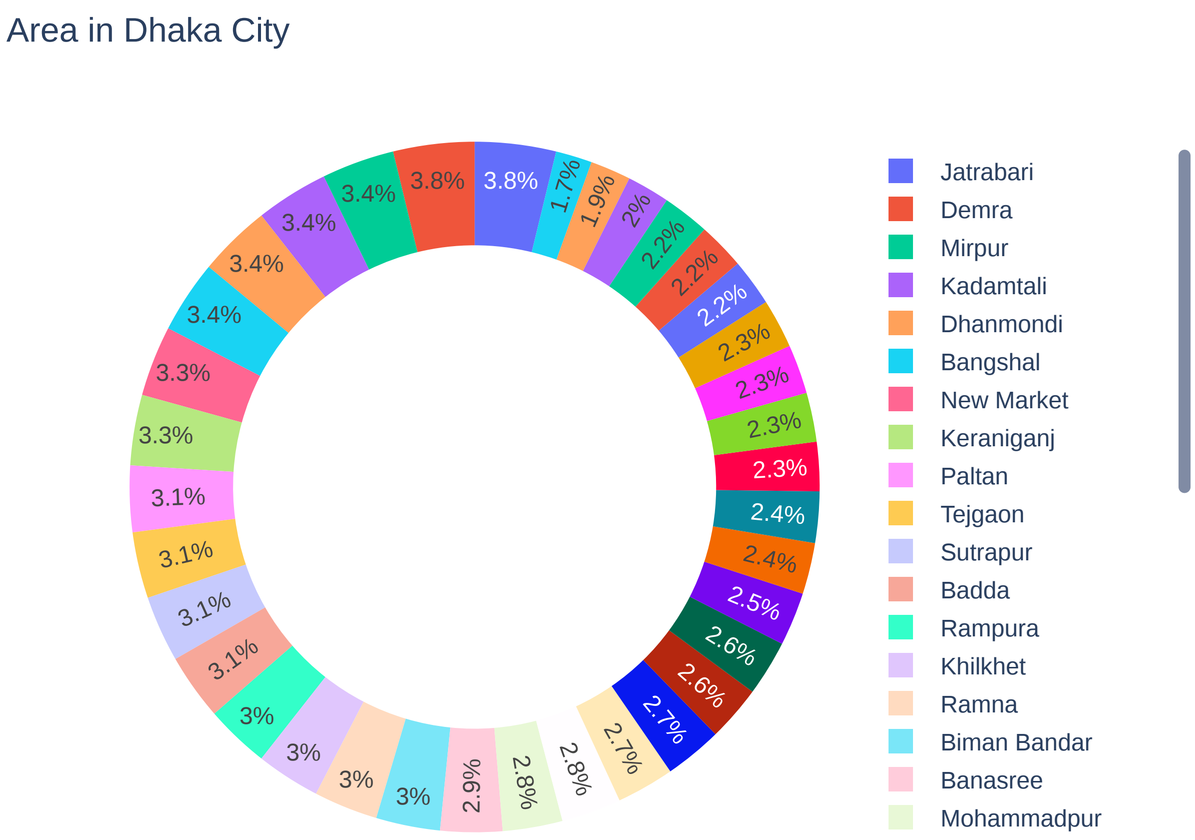
In [40]:

```
Age =df['Age'].value_counts()
transctions = Age.index
quantity =Age.values
figure =px.pie(df,
               values=quantity,
               names =transctions,hole=.70,
               title=" Dhaka City People Age ")

figure.show()
```



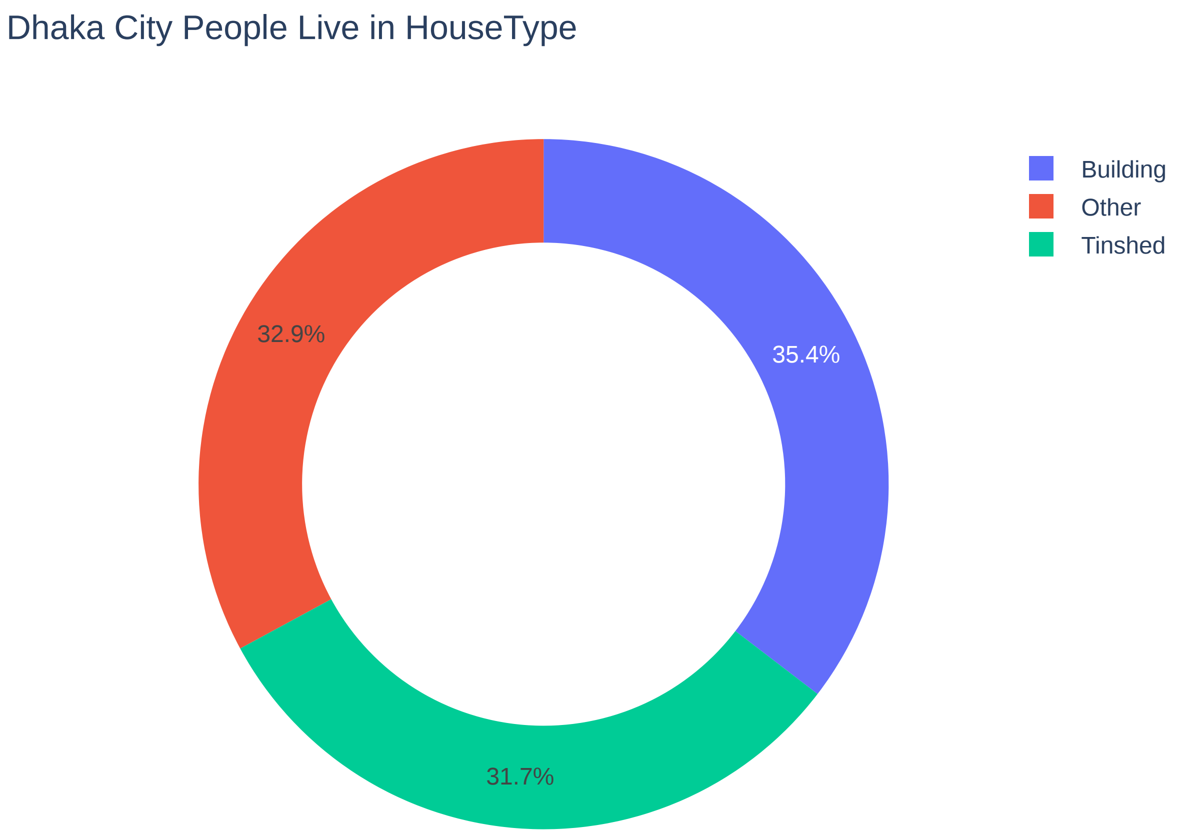
In [41]:

```
 Area =df['Area'].value_counts()
transctions = Area.index
quantity =Area.values
figure =px.pie(df,
               values=quantity,
               names =transctions,hole=.70,
               title="  Area in Dhaka City ")

figure.show()
```



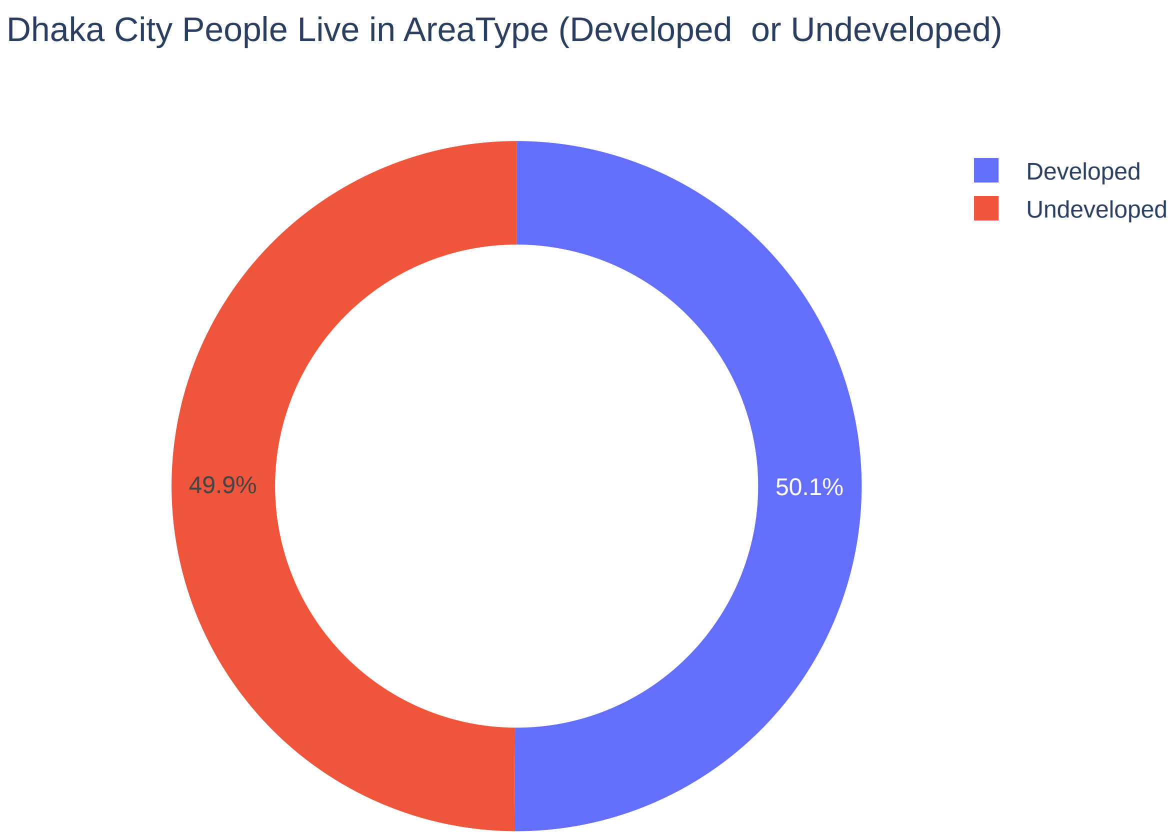
In [42]:

```
 HouseType=df['HouseType'].value_counts()
transctions = HouseType.index
quantity =HouseType.values
figure =px.pie(df,
               values=quantity,
               names =transctions,hole=.70,
               title=" Dhaka City People Live in HouseType ")

figure.show()
```



In [43]:

```
AreaType=df['AreaType'].value_counts()
transctions =AreaType.index
quantity =AreaType.values
figure =px.pie(df,
               values=quantity,
               names =transctions,hole=.70,
               title=" Dhaka City People Live in AreaType (Developed  or Undeveloped)")

figure.show()
```



Pandas Profiling

In [44]:

```
import pandas as pd
from ydata_profiling import ProfileReport
ProfileReport(x, title="Dhaka_City-information_Analysis_Report")
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Dhaka_City-information_Analysis_Report

Overview Variables Interactions Correlations Missing values Sample Duplicate rows

Overview

Overview Alerts 5 Reproduction

Dataset statistics

Number of variables	9
Number of observations	1001
Missing cells	9
Missing cells (%)	0.1%
Duplicate rows	6
Duplicate rows (%)	0.6%
Total size in memory	70.5 KiB
Average record size in memory	72.1 B

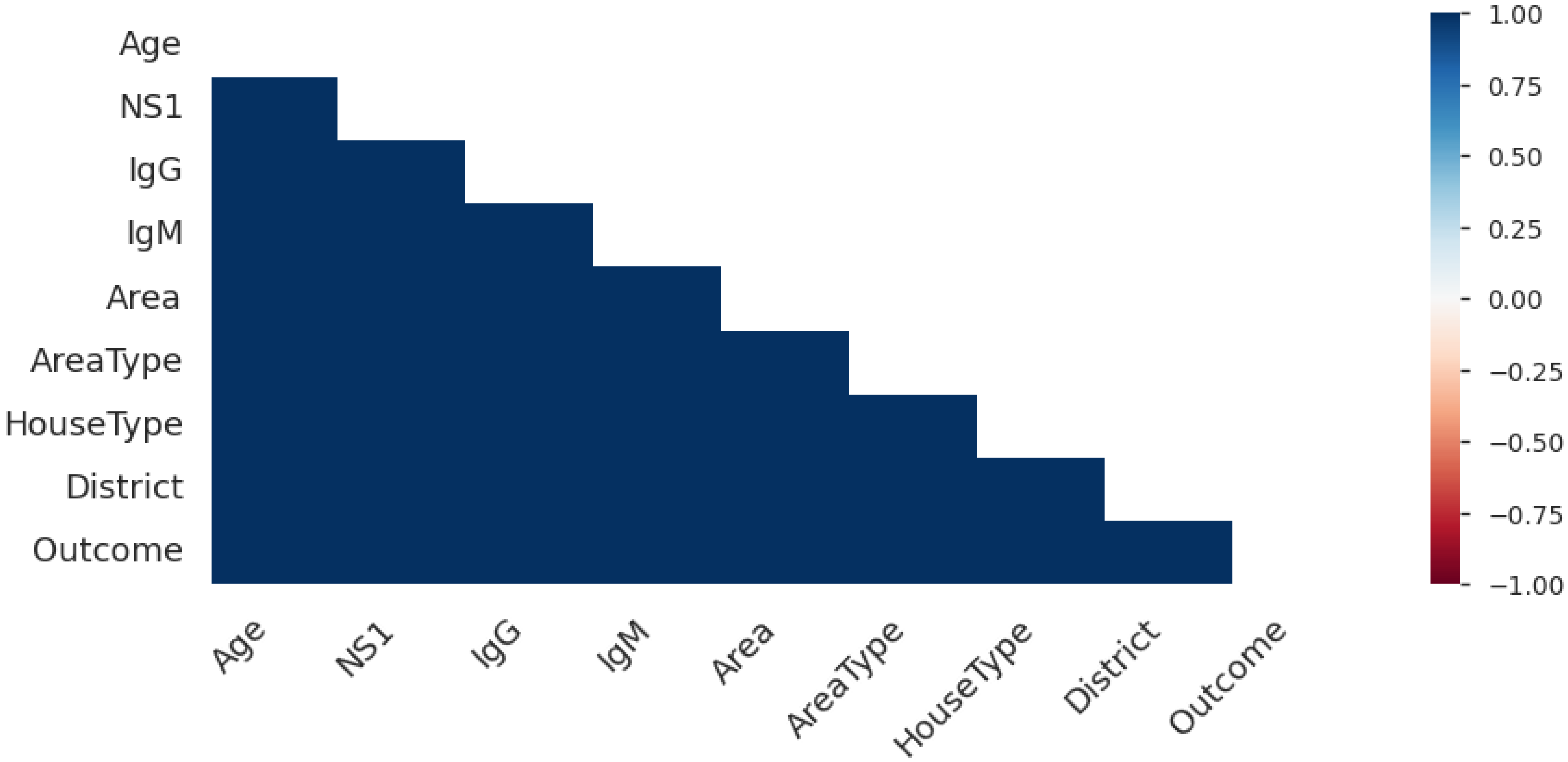
Variable types

Numeric	1
Categorical	8

Variables

Select Columns

```
Out[44]:
```


Age
NS1
IgG
IgM
Area
AreaType
HouseType
District
Outcome
Age
NS1
IgG
IgM
Area
AreaType
HouseType
District
Outcome
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

```
In [ ]:
```