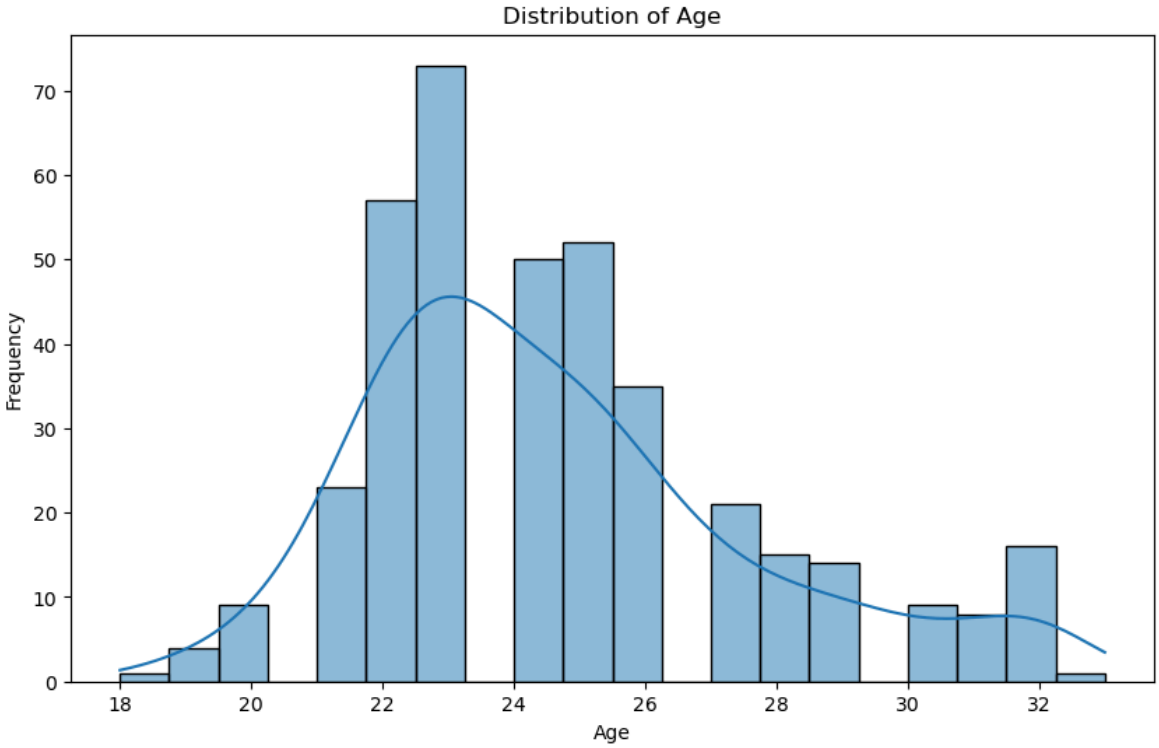
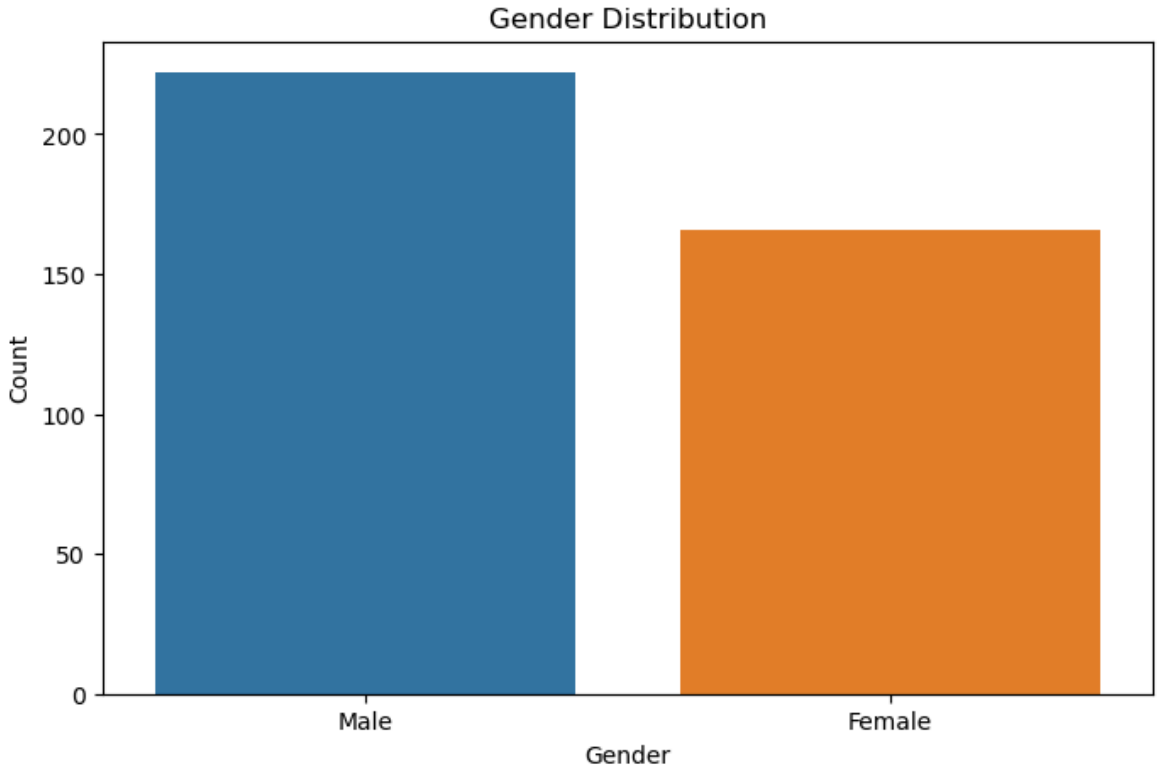
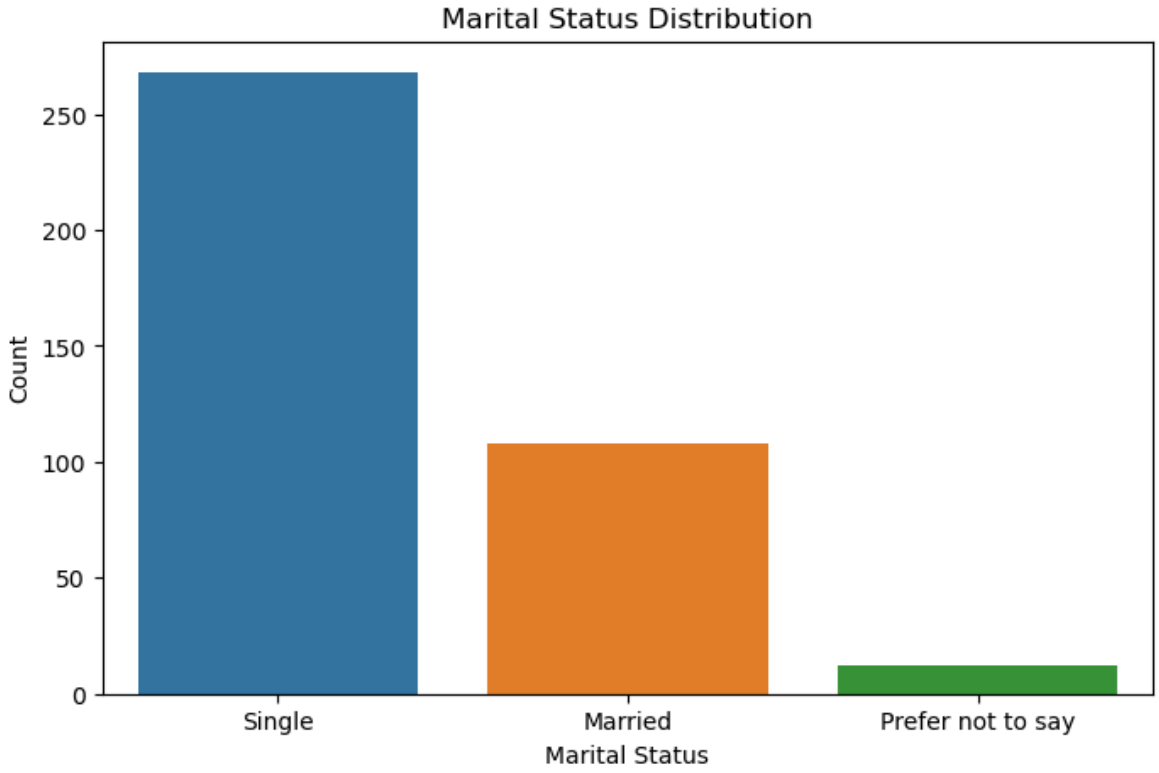
In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv("D:/AI_Course/Assignments/archive/onlinefoods.csv")

# Display basic information about the dataset
print(data.head())  # Display the first few rows
print(data.info())  # Display information about columns and data types

# Summary statistics
print(data.describe())

# EDA using visualizations
# 1. Distribution of Age
plt.figure(figsize=(10, 6))
sns.histplot(data['Age'], bins=20, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

# 2. Gender distribution

gender_counts = data['Gender'].value_counts()

# Plot gender distribution
plt.figure(figsize=(8, 5))
sns.barplot(x=gender_counts.index, y=gender_counts.values)
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()

# 3. Marital Status distribution
# Count the occurrences of each marital status category
marital_counts = data['Marital Status'].value_counts()

# Plot marital status distribution
plt.figure(figsize=(8, 5))
sns.barplot(x=marital_counts.index, y=marital_counts.values)
plt.title('Marital Status Distribution')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.show()

# 4. Occupation distribution
# Count the occurrences of each occupation category
occupation_counts = data['Occupation'].value_counts()

# Plot occupation distribution
plt.figure(figsize=(12, 6))
sns.barplot(x=occupation_counts.index, y=occupation_counts.values)
plt.title('Occupation Distribution')
plt.xlabel('Occupation')
plt.ylabel('Count')
plt.xticks(rotation=45)
```

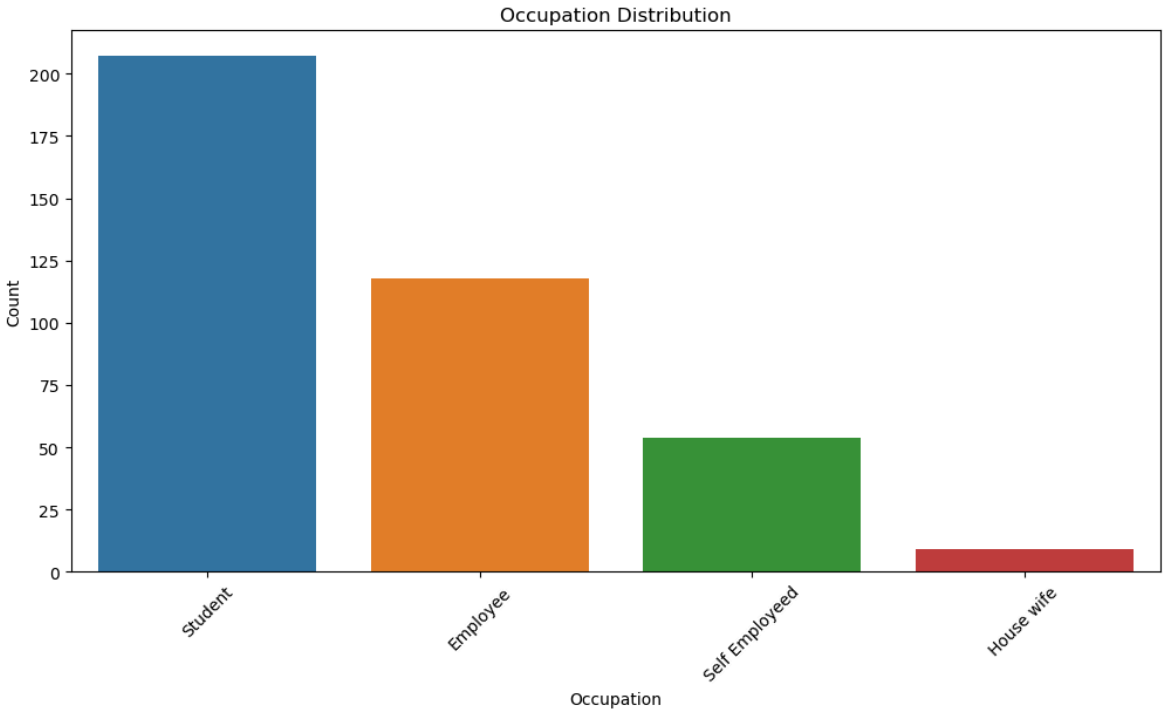
```
plt.show()

# 5. Monthly Income distribution
plt.figure(figsize=(10, 6))
sns.histplot(data['Monthly Income'], bins=20, kde=True)
plt.title('Distribution of Monthly Income')
plt.xlabel('Monthly Income')
plt.ylabel('Frequency')
plt.show()

# 6. Family Size distribution
plt.figure(figsize=(8, 5))
sns.countplot(data['Family size'])
plt.title('Family Size Distribution')
plt.xlabel('Family size')
plt.ylabel('Count')
plt.show()

# 7. Correlation between variables
data_numeric = data.select_dtypes(include=['int64', 'float64'])
plt.figure(figsize=(10, 8))
sns.heatmap(data_numeric.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation between variables')
plt.show()

# 8. Pairplot for selected variables
sns.pairplot(data[['Age', 'Monthly Income', 'Family size']])
plt.show()

# Generate heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(data_numeric.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation between variables')

# Save the plot as a PDF file
download_path = r"D:\AI_Course\Assignments\correlation_heatmap.pdf"

# Save the plot as a PDF file
plt.savefig(download_path)

plt.show()
```

```
   Age  Gender Marital Status Occupation  Monthly Income  \
0   20  Female         Single    Student       No Income
1   24  Female         Single    Student  Below Rs.10000
2   22    Male         Single    Student  Below Rs.10000
3   22  Female         Single    Student       No Income
4   22    Male         Single    Student  Below Rs.10000

  Educational Qualifications  Family size  latitude  longitude  Pin code  \
0              Post Graduate            4   12.9766    77.5993    560001
1                   Graduate            3   12.9770    77.5773    560009
2              Post Graduate            3   12.9551    77.6593    560017
3                   Graduate            6   12.9473    77.5616    560019
4              Post Graduate            4   12.9850    77.5533    560010

  Output   Feedback Unnamed: 12
0    Yes   Positive         Yes
1    Yes   Positive         Yes
2    Yes  Negative          Yes
3    Yes   Positive         Yes
4    Yes   Positive         Yes
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 13 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Age                          388 non-null    int64
 1   Gender                       388 non-null    object
 2   Marital Status               388 non-null    object
 3   Occupation                   388 non-null    object
 4   Monthly Income               388 non-null    object
 5   Educational Qualifications   388 non-null    object
 6   Family size                  388 non-null    int64
 7   latitude                     388 non-null    float64
 8   longitude                    388 non-null    float64
 9   Pin code                     388 non-null    int64
 10  Output                       388 non-null    object
 11  Feedback                     388 non-null    object
 12  Unnamed: 12                  388 non-null    object
dtypes: float64(2), int64(3), object(8)
memory usage: 39.5+ KB
None
              Age  Family size    latitude   longitude       Pin code
count  388.000000   388.000000  388.000000  388.000000     388.000000
mean    24.628866     3.280928   12.972058   77.600160  560040.113402
std      2.975593     1.351025    0.044489    0.051354      31.399609
min     18.000000     1.000000   12.865200   77.484200  560001.000000
25%     23.000000     2.000000   12.936900   77.565275  560010.750000
50%     24.000000     3.000000   12.977000   77.592100  560033.500000
75%     26.000000     4.000000   12.997025   77.630900  560068.000000
max     33.000000     6.000000   13.102000   77.758200  560109.000000
C:\Users\Muqthar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

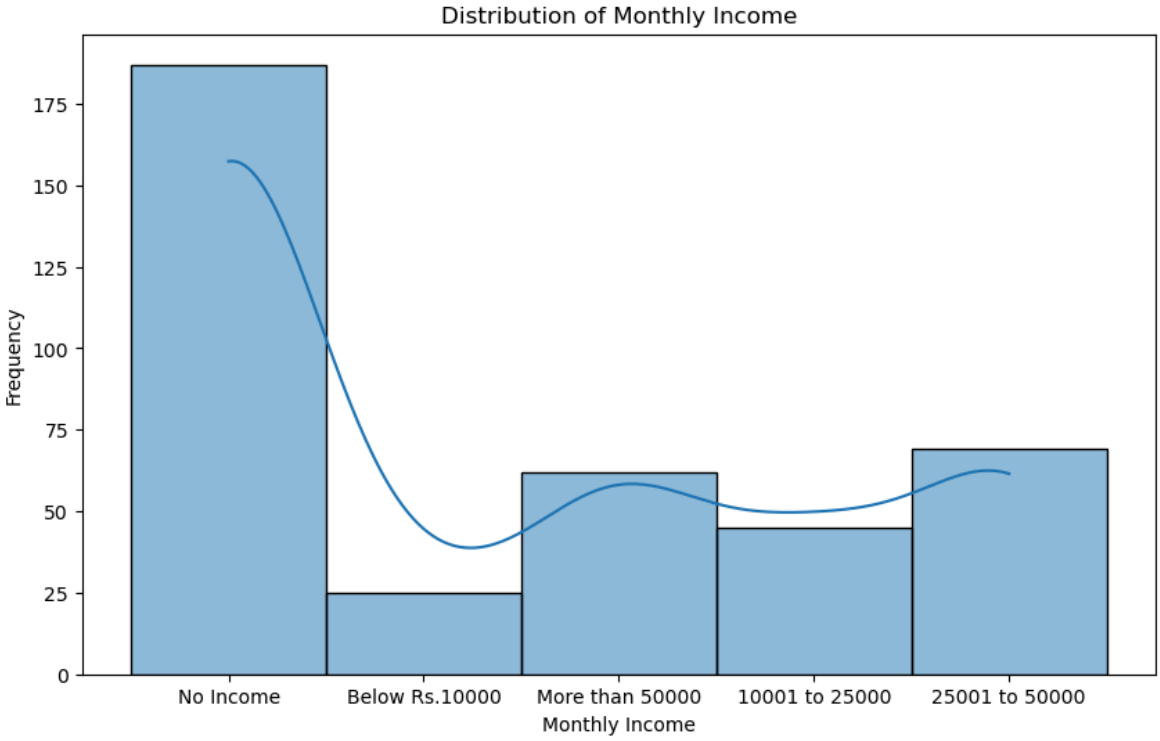

Distribution of Age
Frequency
Age
0
10
20
30
40
50
60
70
18
20
22
24
26
28
30
32

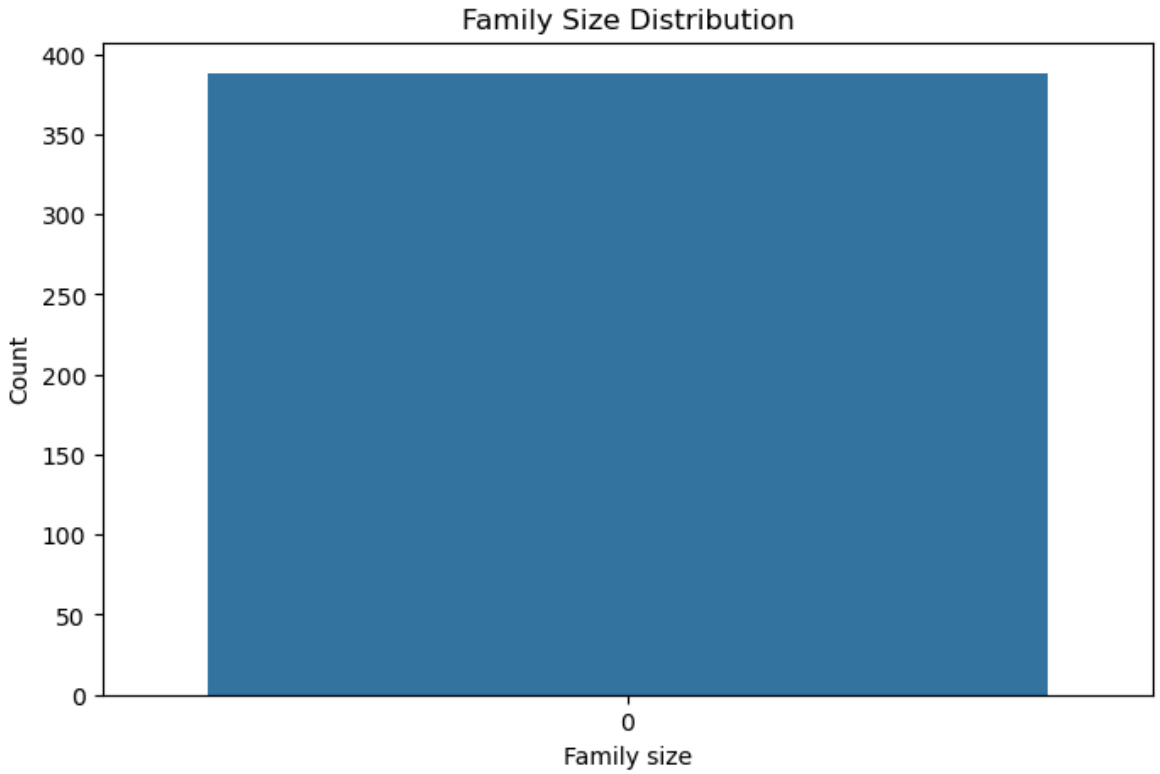

Gender Distribution
Count
Gender
0
50
100
150
200
Male
Female

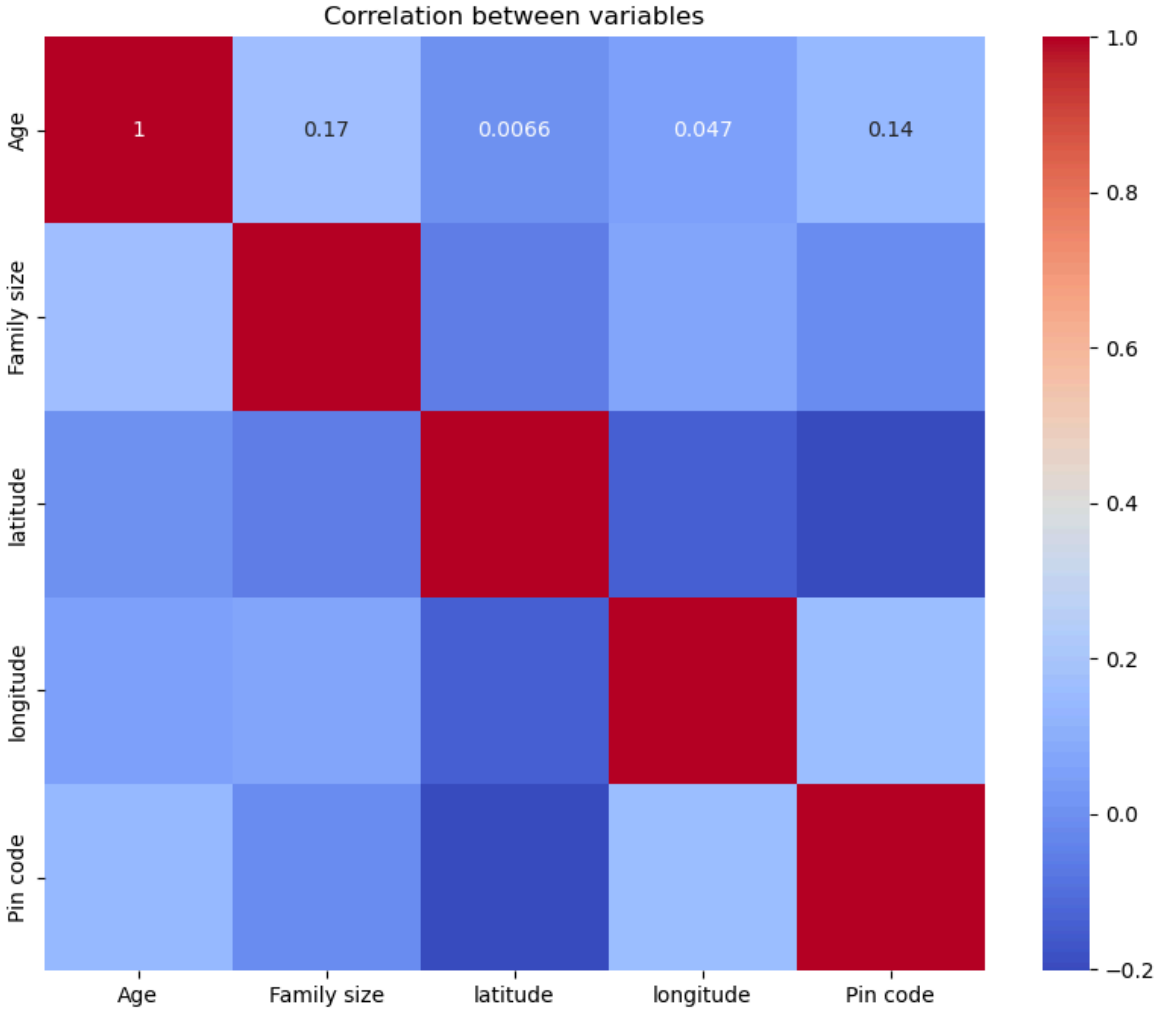




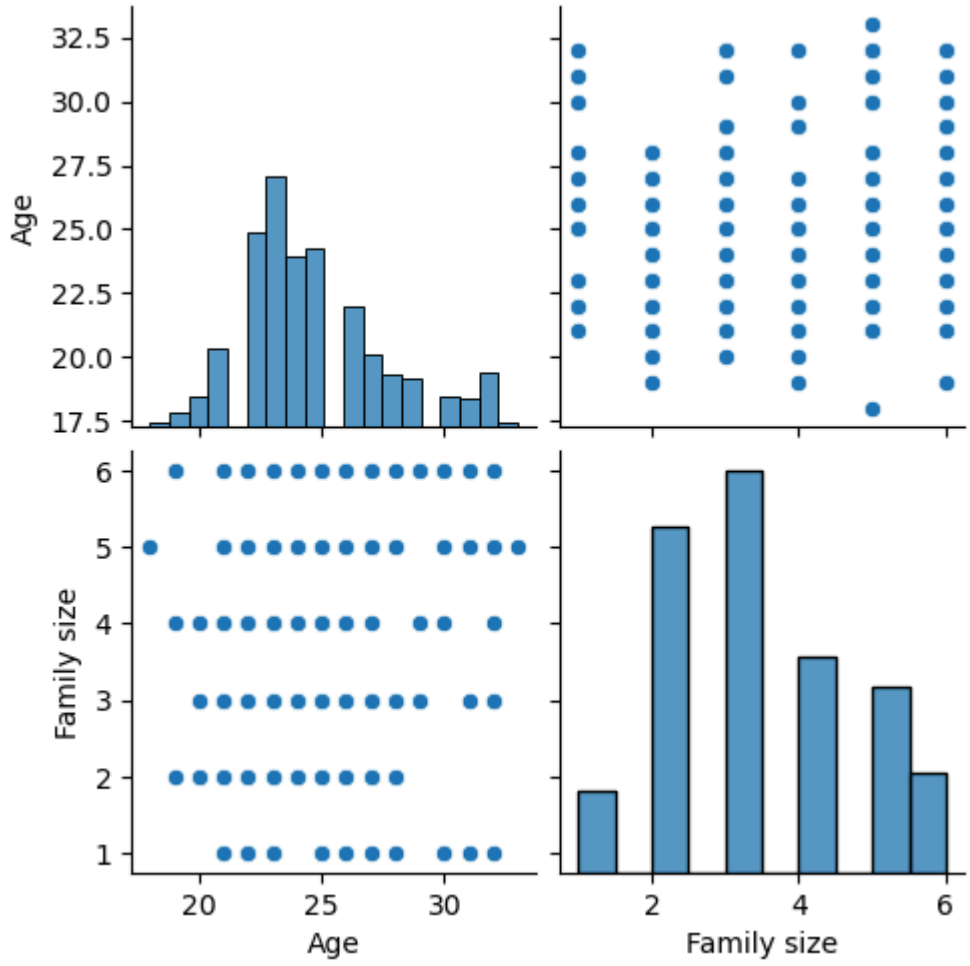

```
C:\Users\Muqthar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

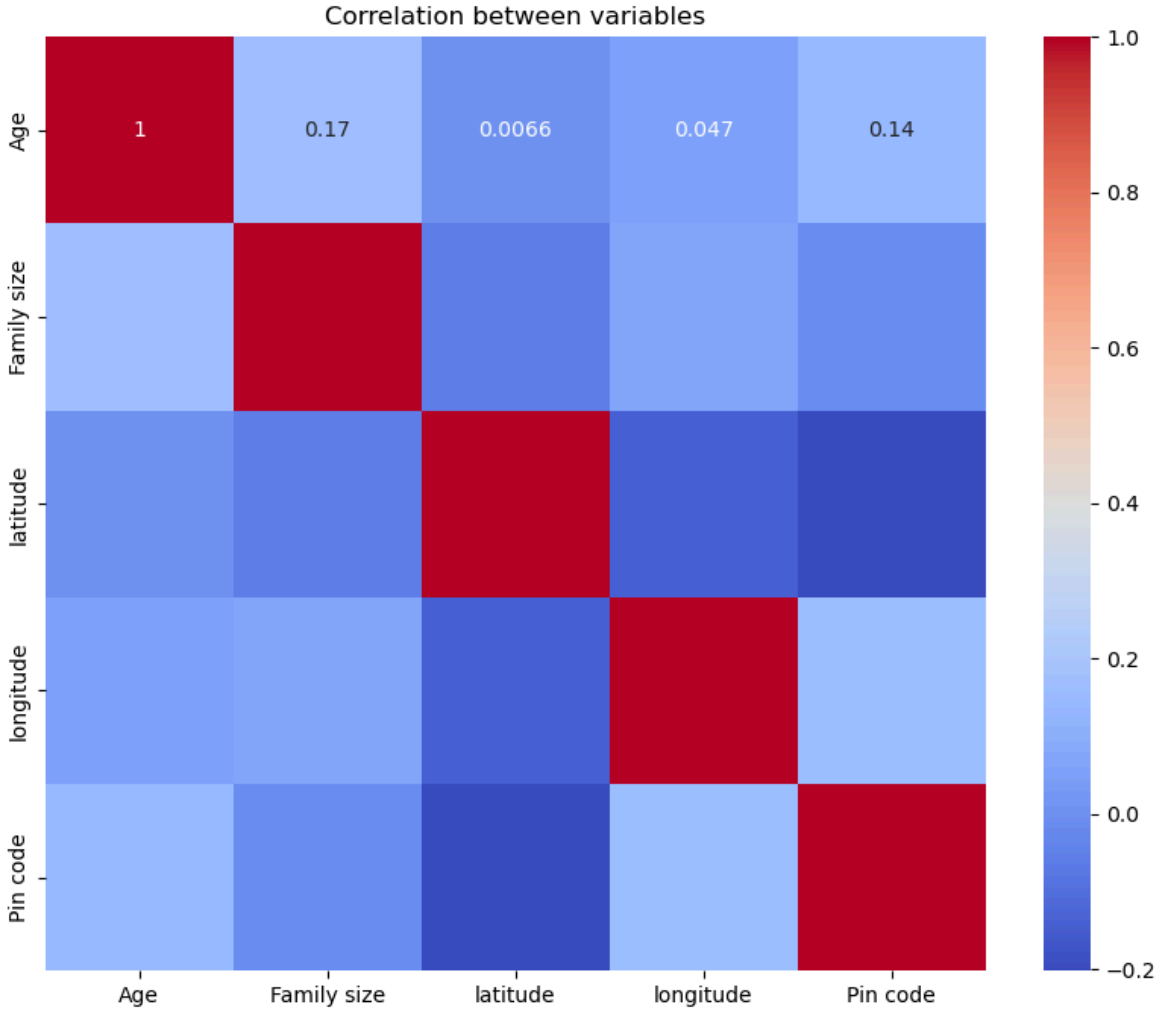

Distribution of Monthly Income
Frequency
175
150
125
100
75
50
25
0
No Income
Below Rs.10000
More than 50000
10001 to 25000
25001 to 50000
Monthly Income


Family Size Distribution
Count
400
350
300
250
200
150
100
50
0
0
Family size




```
C:\Users\Muqthar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Muqthar\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```


Age
Family size
32.5
30.0
27.5
25.0
22.5
20.0
17.5
6
5
4
3
2
1
20
25
30
2
4
6


Correlation between variables
Age
Family size
latitude
longitude
Pin code
1
0.17
0.0066
0.047
0.14
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

In [ ]: