In [31]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.cluster import KMeans


df = pd.read_csv(r"D:\Document\MCA3\Python\activity3\africa_individual_migration.c

df.head(50)
```

Out[31]:

	PersonID	Age	Gender	Country_of_Birth	Destination_Country	Migration_Reason
0	P100000	56	Female	Sao Tome and Principe	Cameroon	Work
1	P100001	43	Female	Djibouti	Central African Republic	Family
2	P100002	34	Other	Morocco	Burundi	Family
3	P100003	0	Male	Burundi	Côte d'Ivoire	Work
4	P100004	51	Female	Mozambique	Congo (Kinshasa)	Family
5	P100005	84	Female	Lesotho	Djibouti	Study
6	P100006	69	Male	Algeria	Tanzania	Conflict
7	P100007	36	Female	Gambia	Eritrea	Work
8	P100008	64	Male	Gambia	Cabo Verde	Work
9	P100009	50	Female	Ghana	Nigeria	Work
10	P100010	0	Female	Madagascar	Morocco	Work
11	P100011	20	Female	Burundi	Mozambique	Work
12	P100012	52	Female	Guinea	Namibia	Work
13	P100013	15	Male	Seychelles	Djibouti	Conflict
14	P100014	60	Female	Djibouti	Cabo Verde	Work
15	P100015	53	Male	Zimbabwe	Guinea	Work
16	P100016	39	Female	Seychelles	Eritrea	Family
17	P100017	25	Female	Burkina Faso	Nigeria	Family
18	P100018	79	Female	Egypt	Comoros	Study
19	P100019	64	Male	Sao Tome and Principe	Somalia	Family
20	P100020	73	Female	Zimbabwe	Togo	Conflict
21	P100021	63	Female	Benin	Uganda	Work
22	P100022	42	Female	Côte d'Ivoire	Namibia	Conflict
23	P100023	16	Male	Senegal	Mali	Work
24	P100024	58	Male	Madagascar	Chad	Work
25	P100025	22	Male	Mozambique	Morocco	Work
26	P100026	6	Female	Niger	Kenya	Work
27	P100027	43	Female	Central African Republic	Mauritania	Study

	PersonID	Age	Gender	Country_of_Birth	Destination_Country	Migration_Reason
28	P100028	52	Other	Cameroon	Chad	Family
29	P100029	70	Male	Liberia	Nigeria	Work
30	P100030	78	Male	Madagascar	Mauritius	Work
31	P100031	33	Female	Algeria	Sierra Leone	Family
32	P100032	45	Female	Morocco	Tanzania	Work
33	P100033	67	Female	Cameroon	Eswatini	Study
34	P100034	77	Female	South Sudan	Equatorial Guinea	Other
35	P100035	77	Male	Mauritania	Cabo Verde	Conflict
36	P100036	15	Female	Sao Tome and Principe	Mauritania	Family
37	P100037	70	Female	Tanzania	Comoros	Study
38	P100038	18	Female	Mauritius	Algeria	Family
39	P100039	1	Male	Cameroon	Guinea	Conflict
40	P100040	9	Male	Ethiopia	Egypt	Work
41	P100041	49	Male	Burundi	South Sudan	Study
42	P100042	80	Female	Tanzania	Morocco	Conflict
43	P100043	11	Male	Malawi	Mozambique	Work
44	P100044	69	Male	Nigeria	Liberia	Other
45	P100045	21	Male	Tanzania	South Sudan	Family
46	P100046	31	Female	Kenya	Seychelles	Family
47	P100047	83	Male	Mauritius	Libya	Work
48	P100048	0	Female	Gambia	Angola	Family
49	P100049	53	Male	Djibouti	Algeria	Work

In [32]:
```
df.tail()
```

Out[32]:

	PersonID	Age	Gender	Country_of_Birth	Destination_Country	Migration_Reason
9995	P109995	24	Male	Algeria	South Sudan	Work
9996	P109996	7	Male	Sudan	Equatorial Guinea	Work
9997	P109997	73	Male	Malawi	Zimbabwe	Work
9998	P109998	31	Male	Togo	Comoros	Work
9999	P109999	16	Female	Egypt	Zimbabwe	Other

In [33]:
```
# Print Age of all migrants from "Chad"
print(df.loc[df['Country_of_Birth'] == 'Chad'])
```

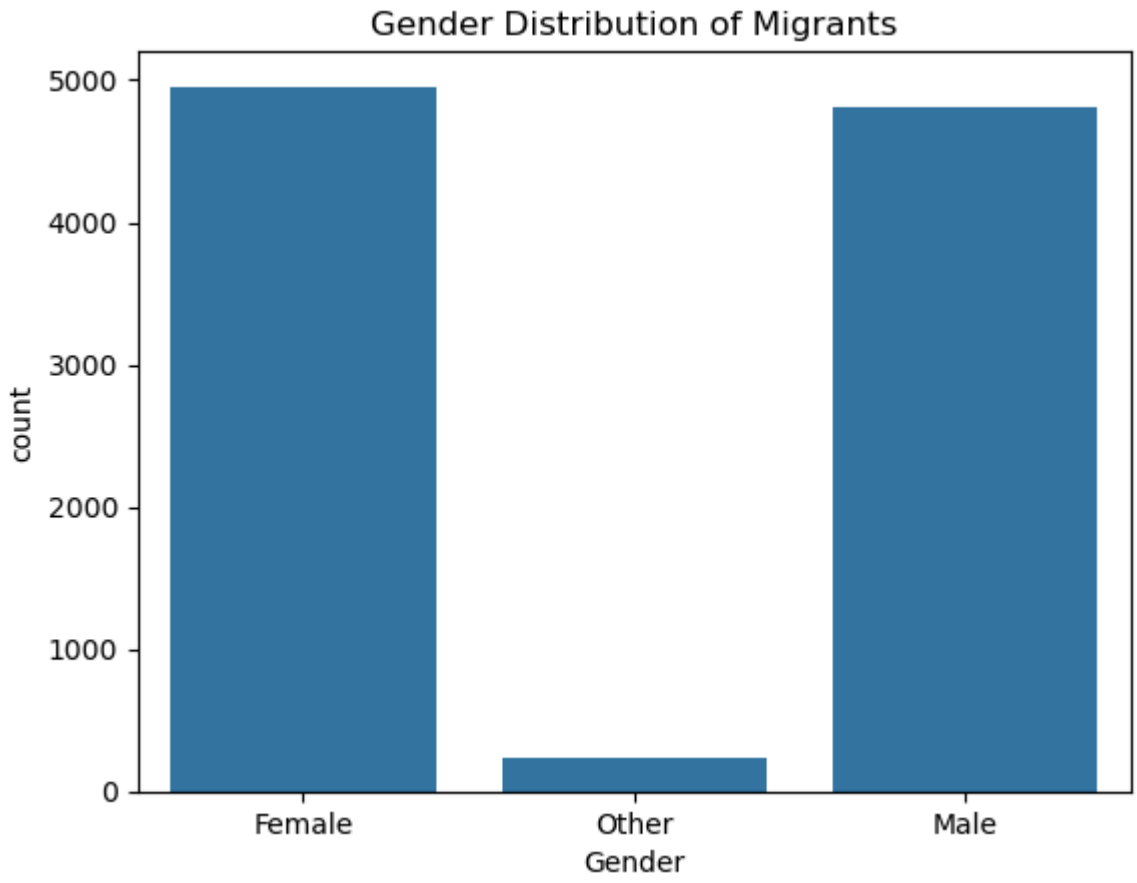
```
     PersonID  Age  Gender Country_of_Birth       Destination_Country  \
87    P100087   40    Male             Chad                   Eritrea
145   P100145   40    Male             Chad                  Djibouti
178   P100178   13    Male             Chad                   Liberia
220   P100220   15  Female             Chad                   Eritrea
262   P100262   32    Male             Chad              Burkina Faso
...       ...  ...     ...              ...                       ...
9440  P109440   87  Female             Chad                     Gabon
9591  P109591   54    Male             Chad                   Tunisia
9772  P109772   83  Female             Chad                   Burundi
9787  P109787   19  Female             Chad  Central African Republic
9897  P109897   26    Male             Chad             Guinea-Bissau

     Migration_Reason      Education_Level  Year_of_Migration
87               Work              Primary               2011
145              Work              Primary               2014
178              Work            Secondary               2005
220          Conflict  No formal education               2007
262             Study             Tertiary               2004
...               ...                  ...                ...
9440             Work            Secondary               2020
9591             Work             Tertiary               2011
9772            Other            Secondary               2010
9787           Family            Secondary               2018
9897         Conflict  No formal education               2000

[203 rows x 8 columns]
```

In [8]:
```
# Basic info
print(df.info())

# Check for missing values
print(df.isnull().sum())

# Check unique values in categorical columns
print(df.nunique())

# Quick statistics
print(df.describe())
```

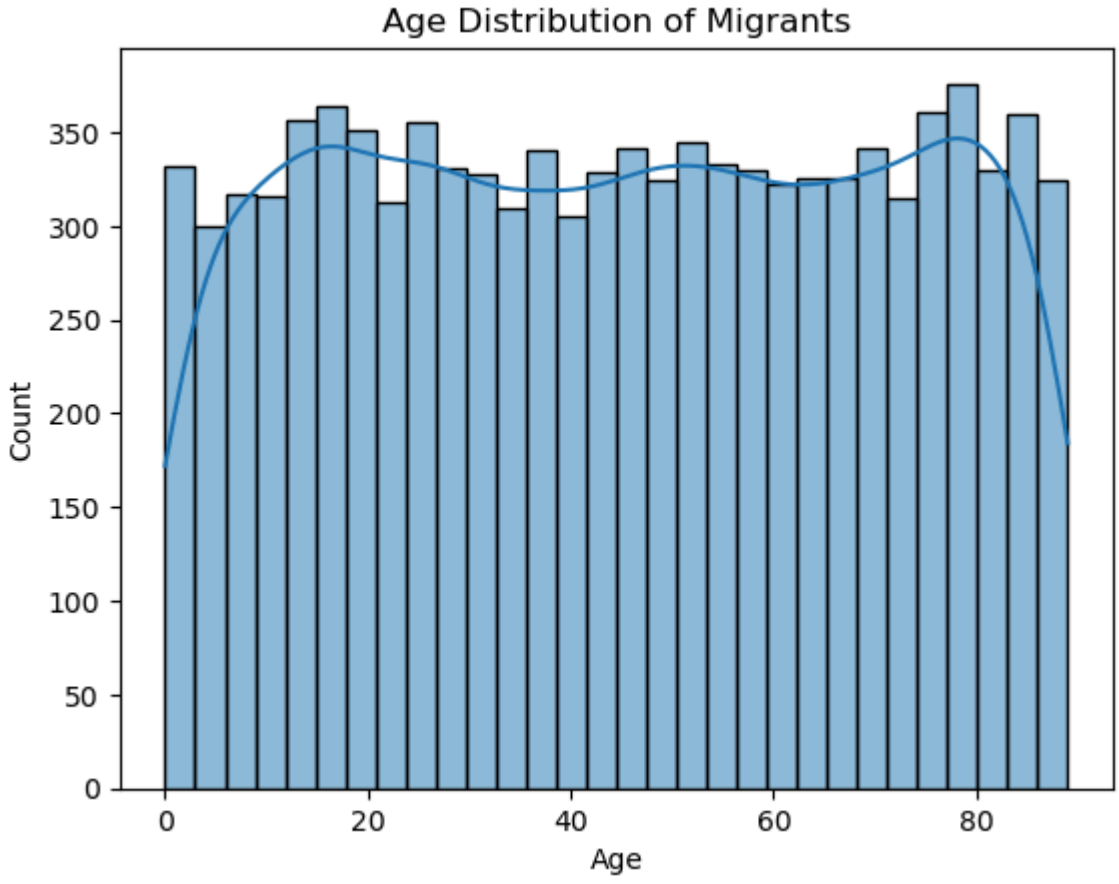
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   PersonID             10000 non-null  object
 1   Age                  10000 non-null  int64 
 2   Gender               10000 non-null  object
 3   Country_of_Birth     10000 non-null  object
 4   Destination_Country  10000 non-null  object
 5   Migration_Reason     10000 non-null  object
 6   Education_Level      10000 non-null  object
 7   Year_of_Migration    10000 non-null  int64 
dtypes: int64(2), object(6)
memory usage: 625.1+ KB
None
PersonID               0
Age                    0
Gender                 0
Country_of_Birth       0
Destination_Country    0
Migration_Reason       0
Education_Level        0
Year_of_Migration      0
dtype: int64
PersonID               10000
Age                       90
Gender                     3
Country_of_Birth          54
Destination_Country       54
Migration_Reason           6
Education_Level            4
Year_of_Migration         25
dtype: int64
                Age  Year_of_Migration
count  10000.000000        10000.00000
mean      44.830700         2012.11730
std       26.035914            7.15215
min        0.000000         2000.00000
25%       22.000000         2006.00000
50%       45.000000         2012.00000
75%       68.000000         2018.00000
max       89.000000         2024.00000
```

In [9]:
```python
#Handle Missing Values
# Fill numeric missing values with mean
numeric_cols = df.select_dtypes(include=['int64','float64']).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# Fill categorical missing values with mode
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])
```

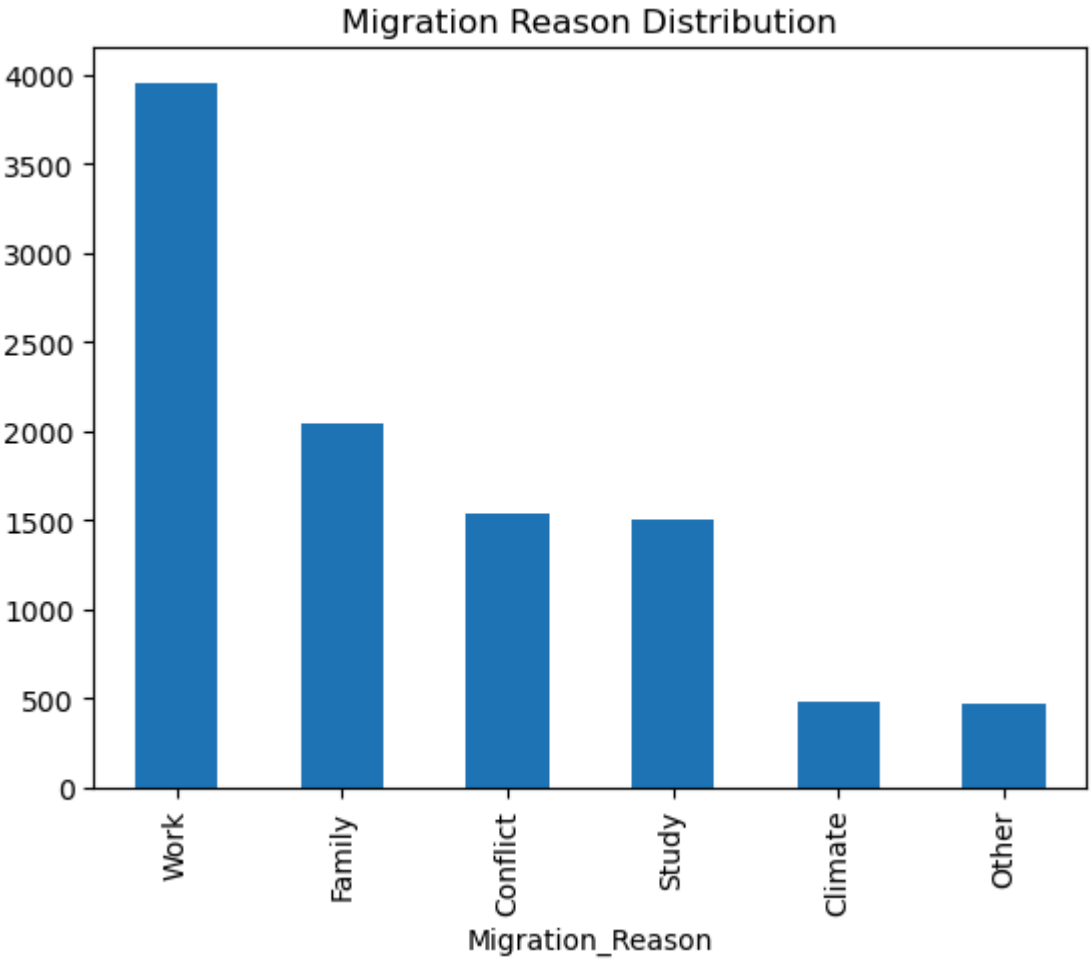
In [11]:
```python
#Exploratory Data Analysis:
# 1. Gender Distribution
sns.countplot(data=df, x='Gender')
plt.title("Gender Distribution of Migrants")
plt.show()
```



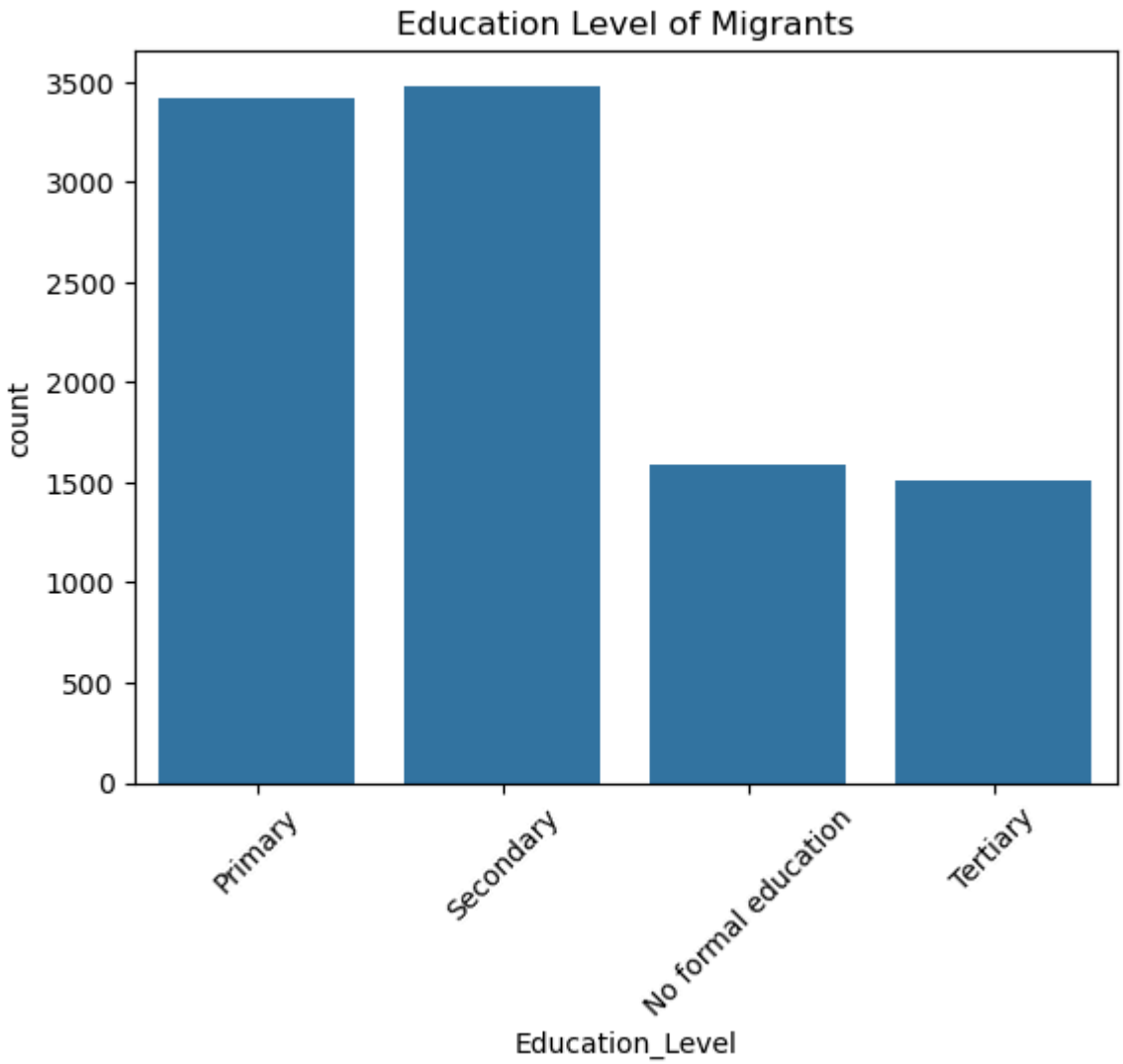
In [13]:
```
#2. Age Distribution
sns.histplot(df['Age'], bins=30, kde=True)
plt.title("Age Distribution of Migrants")
plt.show()
```



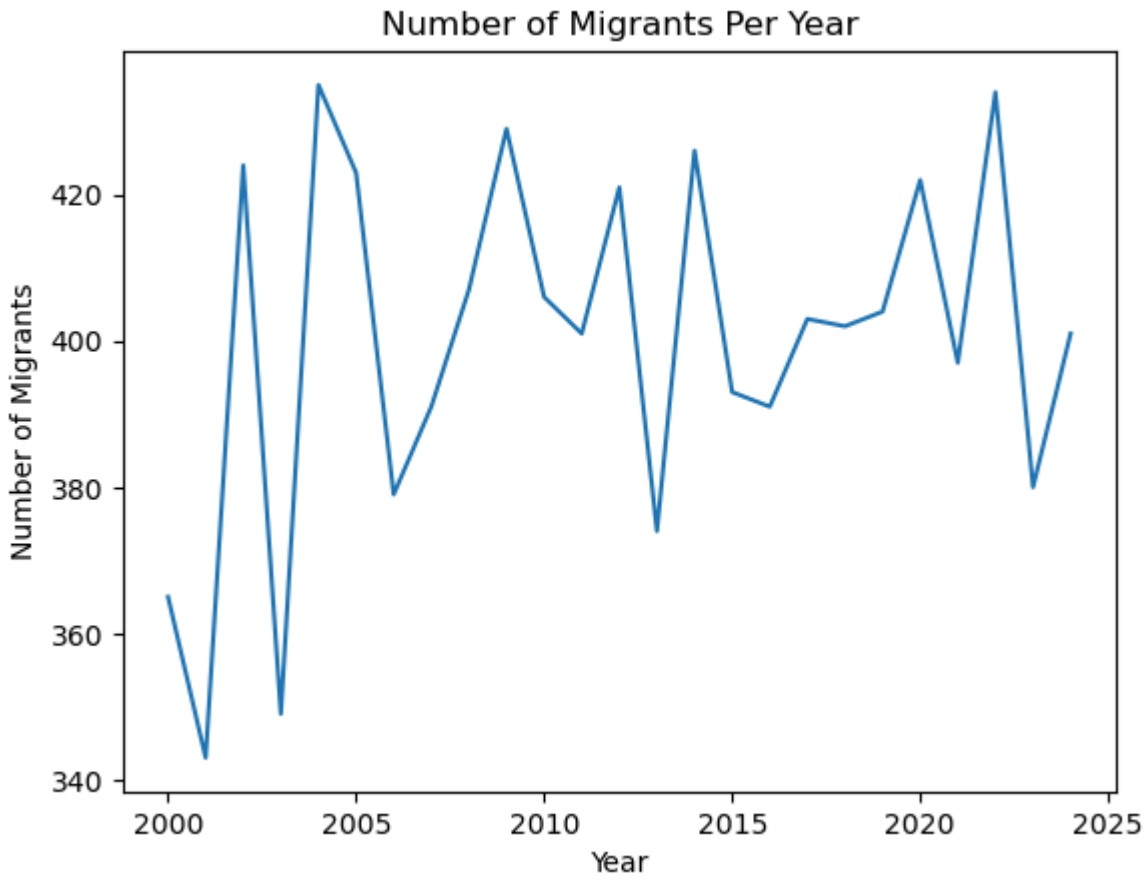
In [15]:
```
#3. Migration Reason
df['Migration_Reason'].value_counts().plot(kind='bar')
plt.title("Migration Reason Distribution")
plt.show()
```



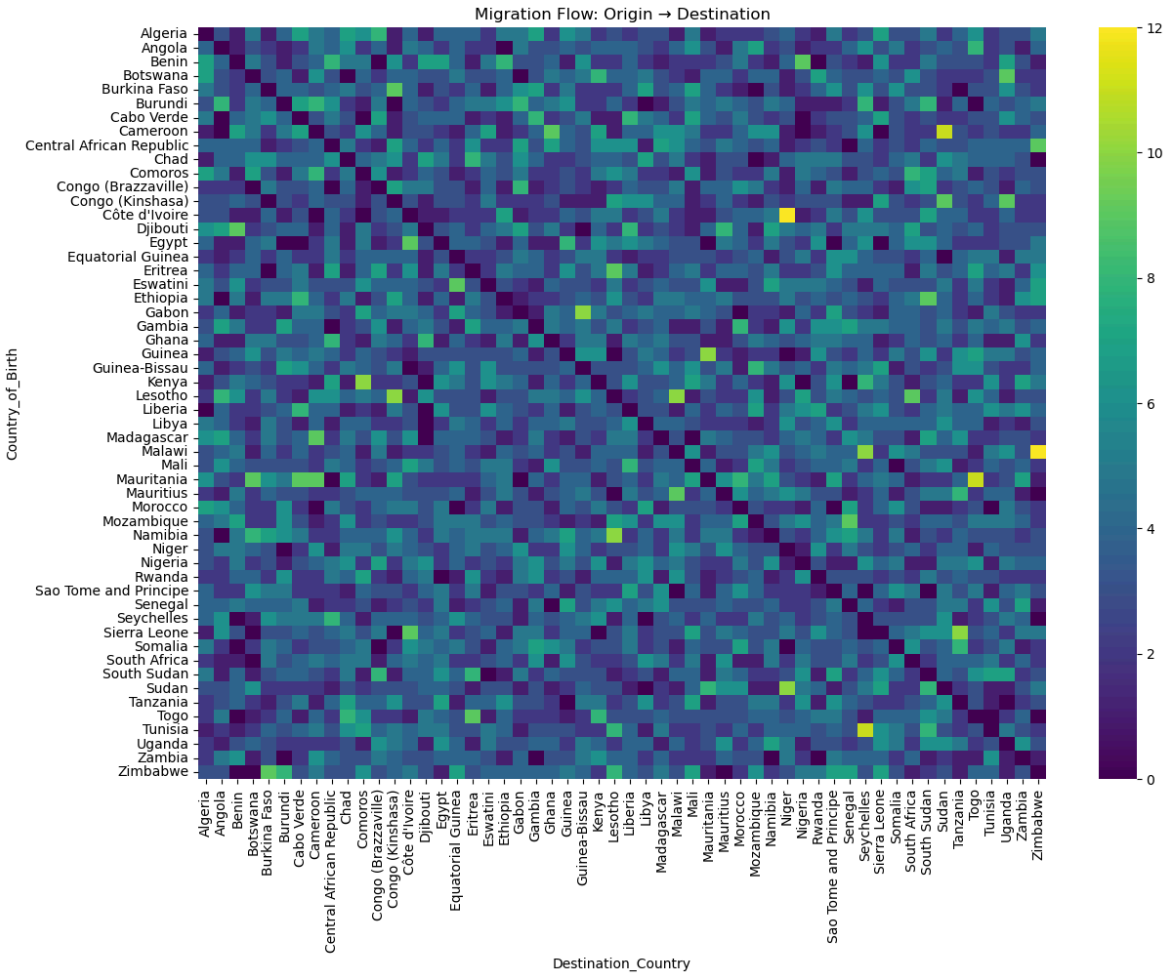
In [17]:
```
# 4. Education Level
sns.countplot(data=df, x='Education_Level')
plt.title("Education Level of Migrants")
plt.xticks(rotation=45)
plt.show()
```



In [19]:
```
#5. Year of Migration Trend
df.groupby('Year_of_Migration').size().plot()
plt.title("Number of Migrants Per Year")
plt.xlabel("Year")
plt.ylabel("Number of Migrants")
plt.show()
```



In [2]:
```
# 6. Origin vs Destination Heatmap
pivot = df.pivot_table(index='Country_of_Birth',
                       columns='Destination_Country',
                       aggfunc='size',
                       fill_value=0)

plt.figure(figsize=(14,10))
sns.heatmap(pivot, cmap="viridis")
plt.title("Migration Flow: Origin → Destination")
plt.show()
```



In [5]:
```python
# 1. Select features & label
X = df[['Age', 'Gender', 'Country_of_Birth',
        'Destination_Country', 'Education_Level']]
y = df['Migration_Reason']   # Target variable
```

In [6]:
```python
# 2. Encode categorical data
label = LabelEncoder()

df['Gender'] = label.fit_transform(df['Gender'])
df['Country_of_Birth'] = label.fit_transform(df['Country_of_Birth'])
df['Destination_Country'] = label.fit_transform(df['Destination_Country'])
df['Education_Level'] = label.fit_transform(df['Education_Level'])
df['Migration_Reason'] = label.fit_transform(df['Migration_Reason'])

X = df[['Age', 'Gender', 'Country_of_Birth',
        'Destination_Country', 'Education_Level']]
y = df['Migration_Reason']
```

In [8]:
```python
# 3. Train-test split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# 4. Scale features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [10]:
```
# 5. Random Forest Classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

# 6. Predict
y_pred = model.predict(X_test)

# 7. Evaluate
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.318
              precision    recall  f1-score   support

           0       0.08      0.02      0.03        93
           1       0.16      0.09      0.11       314
           2       0.19      0.18      0.19       391
           3       0.08      0.01      0.02        91
           4       0.11      0.06      0.08       289
           5       0.41      0.63      0.49       822

    accuracy                           0.32      2000
   macro avg       0.17      0.16      0.15      2000
weighted avg       0.25      0.32      0.27      2000
```

In [12]:
```
#CLUSTERING
df_cluster = df[['Age', 'Gender', 'Education_Level']]

# Encode education level
df_cluster = df_cluster.copy()
df_cluster['Education_Level'] = label.fit_transform(df_cluster['Education_Level'])

# Scale
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_cluster)
```

In [15]:
```
# KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)

print(df[['Age', 'Gender', 'Education_Level', 'Cluster']].head())
```

```
   Age  Gender  Education_Level  Cluster
0   56       0                1        1
1   43       0                2        2
2   34       2                0        0
3    0       1                2        0
4   51       0                2        2
```

In [16]:
```
#REGRESSION
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

X = df[['Age', 'Gender']]
y = df['Year_of_Migration']
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("MSE:", mean_squared_error(y_test, y_pred))
```

```
MSE: 53.301074211508826
```