Team 4 Data Analysis

Entrée [1]:

```
import pandas as pd

df = pd.read_csv("msamd_14454.csv")
df.head()
```

```
/var/folders/lj/fby1hwmn6vj5h02z5j0952hw0000gn/T/ipykernel_16762/41044
24442.py:3: DtypeWarning: Columns (22,23,24,26,27,28,29,30,31,32,33,3
8,43,44) have mixed types. Specify dtype option on import or set low_m
emory=False.
  df = pd.read_csv("msamd_14454.csv")
```

Out[1]:

	activity_year	lei	derived_msa-md	state_code	county_code	cens
0	2021	549300DAUXQ2DCY4H838	14454	MA	25021.0	2.502
1	2021	549300DAUXQ2DCY4H838	14454	MA	25021.0	2.502
2	2021	549300DAUXQ2DCY4H838	14454	MA	25023.0	2.502
3	2021	549300DAUXQ2DCY4H838	14454	MA	25023.0	2.502
4	2021	549300DAUXQ2DCY4H838	14454	MA	25025.0	2.502

5 rows × 99 columns

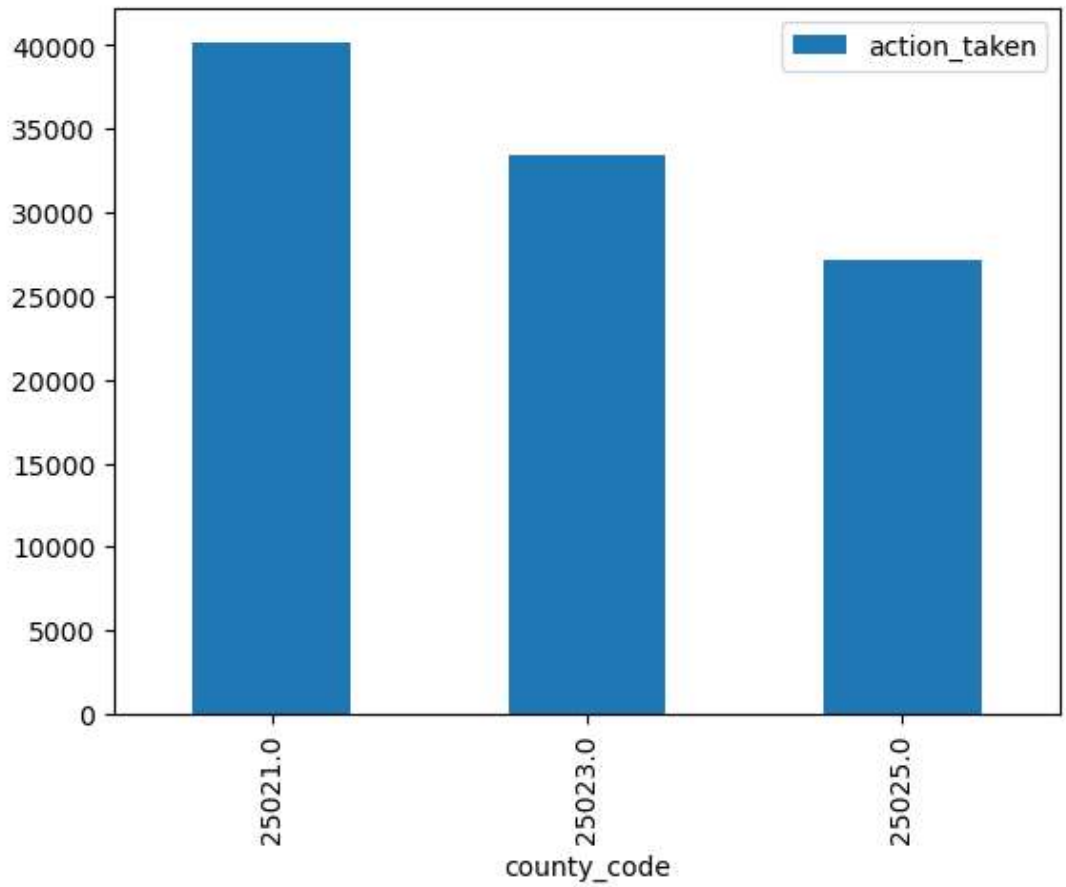
By county determine how many loans were taken.

A loan being taken has a value of 1 in the 'action_taken' column

Entrée [2]:

```
df3 = df[['county_code', 'action_taken']]
df3['action_taken'].map({1: 1, 2: 0}).fillna(0)
df3 = df3.query('action_taken == 1')
df3.groupby('county_code').count().plot(kind='bar')
```

Out[2]: <AxesSubplot:xlabel='county_code'>



Entrée [3]:

```
county_codes_dict = {25021: 'Norfolk', 25023: 'Plymouth', 25025: 'Suff
per_county_count = df3['county_code'].value_counts()

for county_code in county_codes_dict.keys():
    print(f"{county_codes_dict[county_code]}: {per_county_count[county
```

```
Norfolk: 40175
Plymouth: 33453
Suffolk: 27140
```

Establish who is participating in first time home ownership programs

Entrée [4]:

```
df2 = df[['applicant_age', 'derived_dwelling_category', 'derived_race'
df2 = df2.loc[((df['applicant_age'] == '25-34') |
               (df['applicant_age'] == '<25')) & ((df['derived_dwellir
                (df['derived_dwelling_category'] == 'Single Family (1-
df2.head(20)
```

Out[4]:

	applicant_age	derived_dwelling_category	derived_race	income
3	25-34	Single Family (1-4 Units):Site-Built	Black or African American	104.0
4	25-34	Single Family (1-4 Units):Site-Built	Race Not Available	86.0
12	25-34	Single Family (1-4 Units):Site-Built	Black or African American	49.0
15	25-34	Single Family (1-4 Units):Site-Built	Black or African American	131.0
16	25-34	Single Family (1-4 Units):Site-Built	Black or African American	NaN
26	25-34	Single Family (1-4 Units):Site-Built	White	174.0
33	25-34	Single Family (1-4 Units):Site-Built	White	110.0
34	25-34	Single Family (1-4 Units):Site-Built	Black or African American	177.0
37	25-34	Single Family (1-4 Units):Site-Built	White	75.0
40	25-34	Single Family (1-4 Units):Site-Built	Black or African American	88.0
44	25-34	Single Family (1-4 Units):Site-Built	White	78.0
47	25-34	Single Family (1-4 Units):Site-Built	Asian	88.0
60	25-34	Single Family (1-4 Units):Site-Built	White	190.0
65	25-34	Single Family (1-4 Units):Site-Built	White	105.0
67	25-34	Single Family (1-4 Units):Site-Built	Race Not Available	177.0
69	25-34	Single Family (1-4 Units):Site-Built	White	143.0
75	25-34	Single Family (1-4 Units):Site-Built	White	124.0
78	25-34	Single Family (1-4 Units):Site-Built	Black or African American	NaN
81	25-34	Single Family (1-4 Units):Site-Built	Asian	87.0
85	25-34	Single Family (1-4 Units):Site-Built	White	215.0

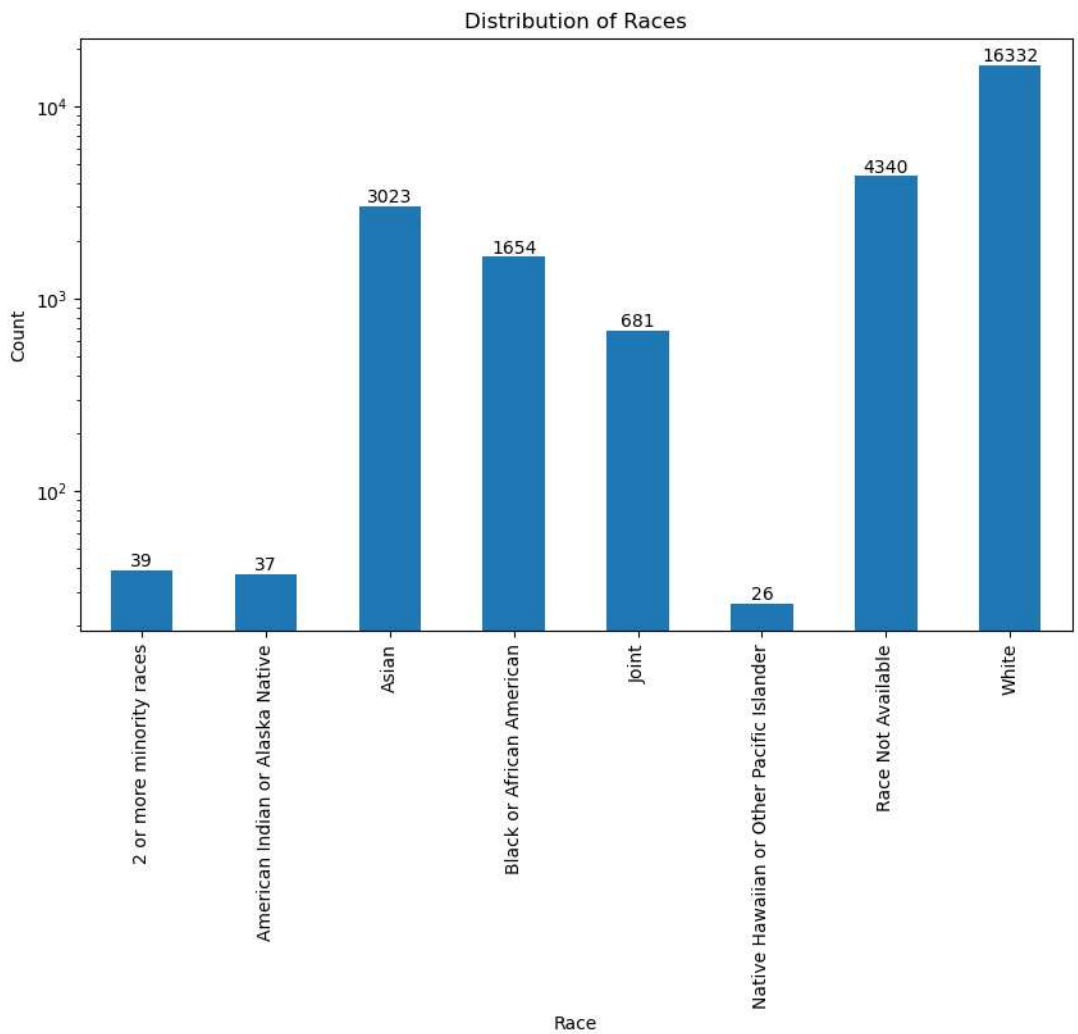
Entrée [5]:

```
import matplotlib.pyplot as plt
# Drop rows with missing income
df2 = df2.dropna(subset=['income'])
df2 = df2.drop(df2[df2['income'] < 0].index)
#print(df2.isnull().sum())
```

Distribution of different races

Entrée [6]:

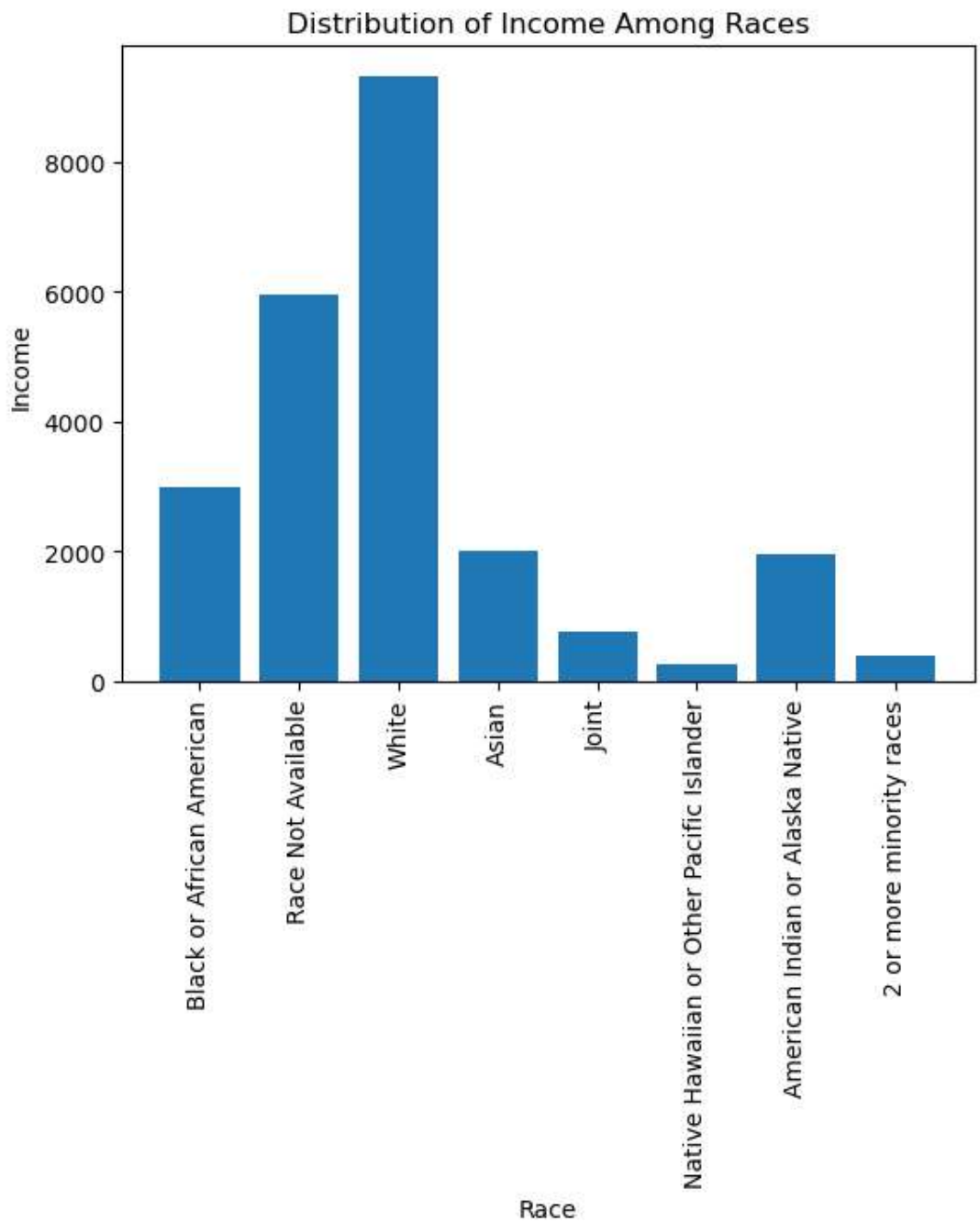
```
race = df2.groupby('derived_race')['derived_race'].count()

plt.figure(figsize=(10, 6))
race.plot(kind='bar')
plt.xlabel('Race')
plt.ylabel('Count')
plt.title('Distribution of Races')

plt.yscale('log')

for i, v in enumerate(race):
    plt.annotate(str(v), xy=(i, v), ha='center', va='bottom')

plt.show()
```



The distribution of income for different derived race.

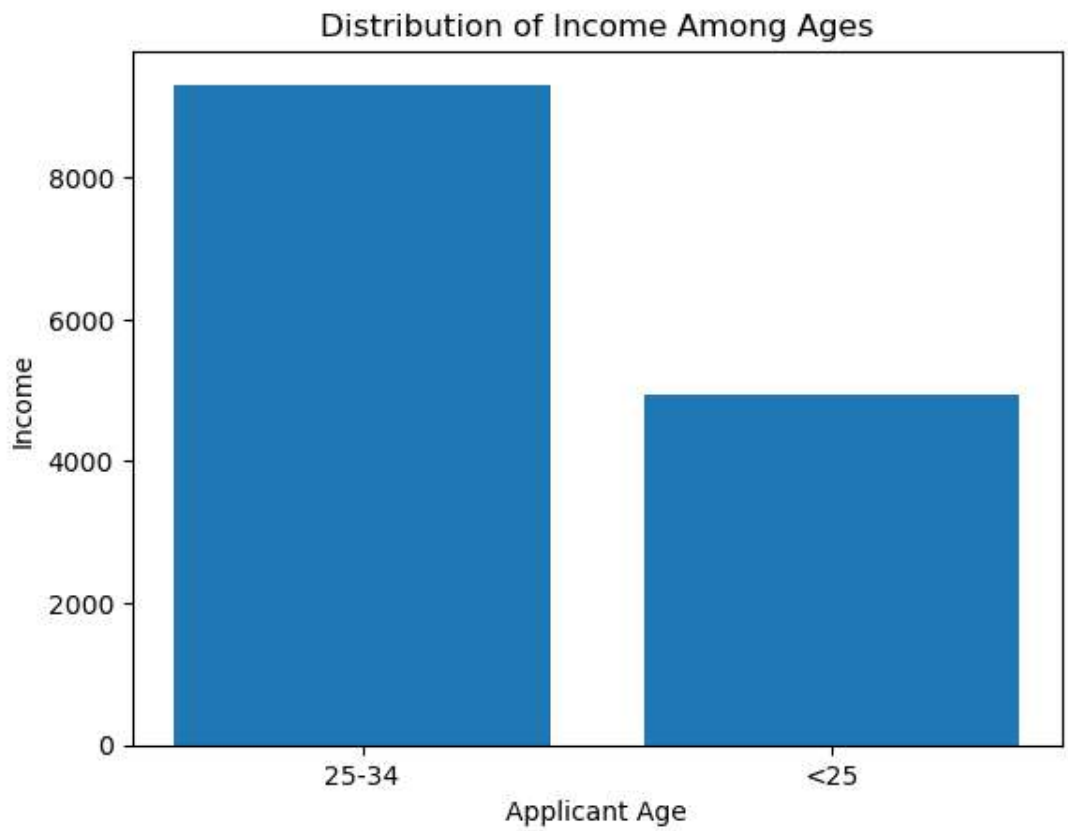
Entrée [7]:

```
plt.bar(df2['derived_race'], df2['income'])
plt.xticks(rotation=90)
plt.xlabel('Race')
plt.ylabel('Income')
plt.title('Distribution of Income Among Races')
plt.show()
```



The distribution for applicants' age and income.

Entrée [8]:
```
plt.bar(df2['applicant_age'], df2['income'])
plt.xlabel('Applicant Age')
plt.ylabel('Income')
plt.title('Distribution of Income Among Ages')
plt.show()
```



Average Income by Housing Type: Accepted

Entrée [9]:
```
new_df=df.groupby('derived_dwelling_category')
```

Entrée [10]:
```
df['denial_reason-1'].unique()
```

Out[10]:
```
array([  10,    9,    3,    1,    4,    2,    5,    7,    6, 1111,
       8])
```

Entrée [11]:
```
accepted = df.loc[df['denial_reason-1']==10]
```

Entrée [12]:

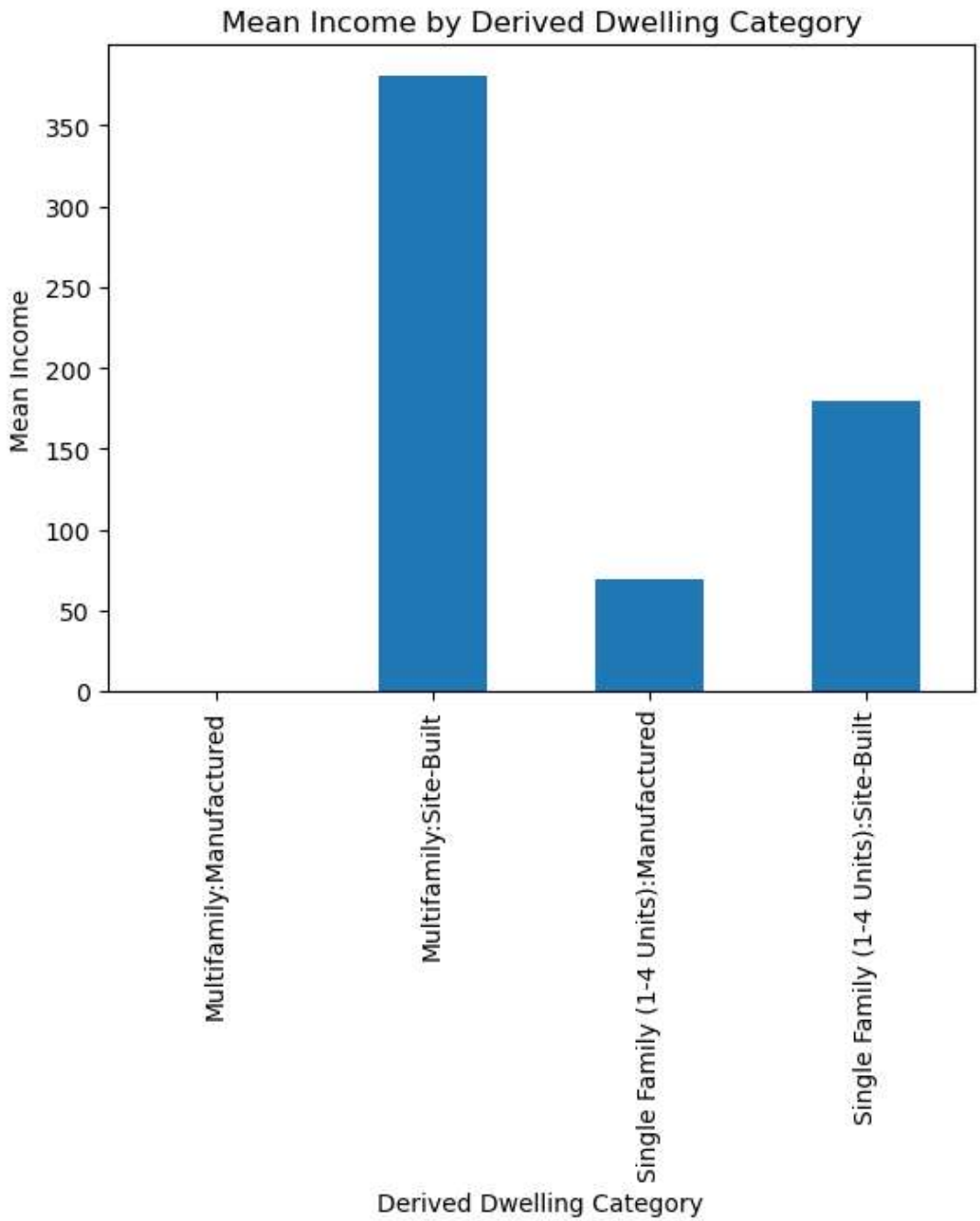
```
rejected = df.loc[df['denial_reason-1']!=10]
```

Entrée [13]:

```
accepted = accepted.groupby('derived_dwelling_category')['income'].mea
```

Entrée [14]:

```
import matplotlib.pyplot as plt

accepted.plot(kind='bar')
plt.xlabel('Derived Dwelling Category')
plt.ylabel('Mean Income')
plt.title('Mean Income by Derived Dwelling Category: Accepted')
plt.show()
```



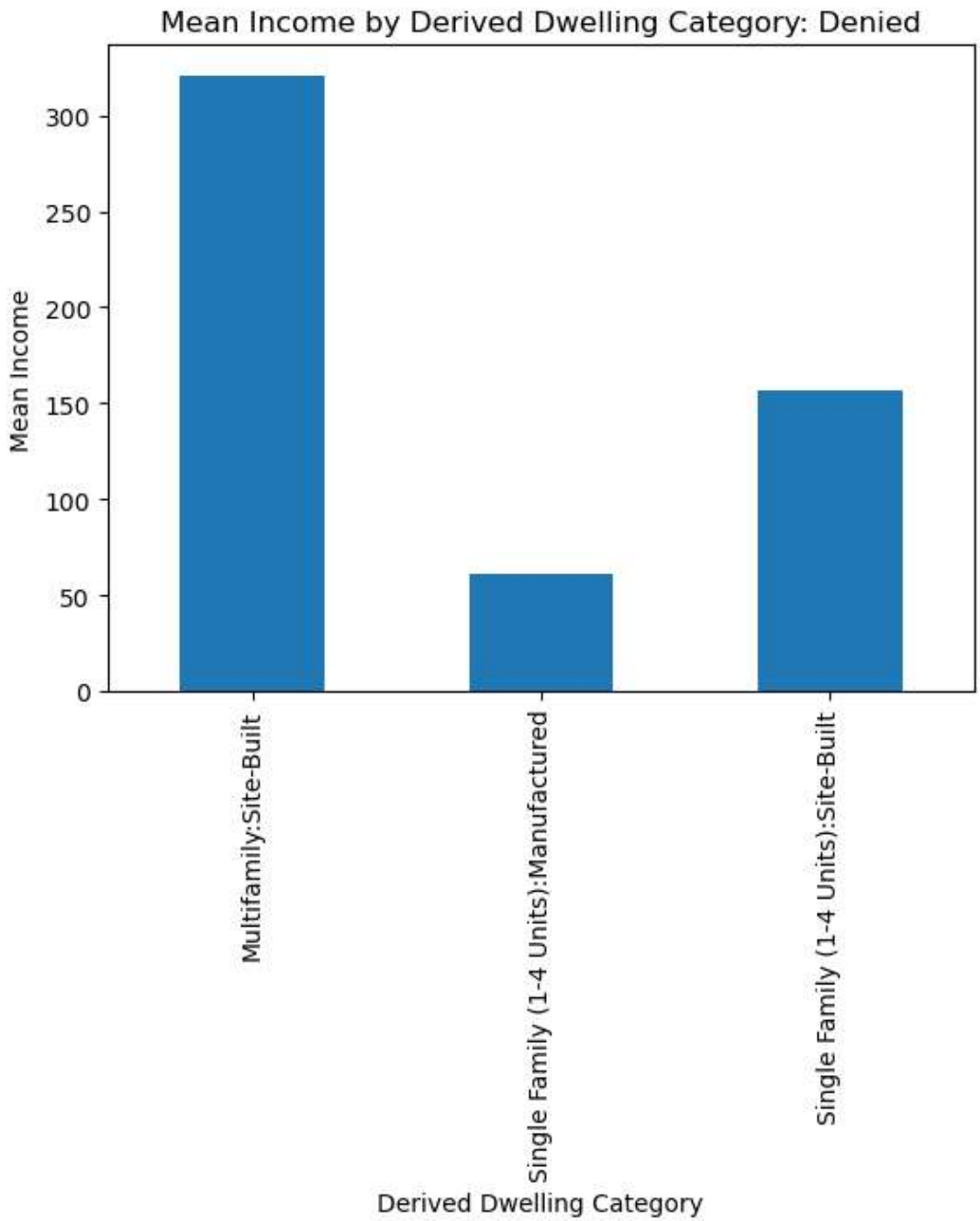
Average Income by Housing Type: Denied

Entrée [16]:

```
rejected = rejected.groupby('derived_dwelling_category')['income'].mea
```

Entrée [22]:

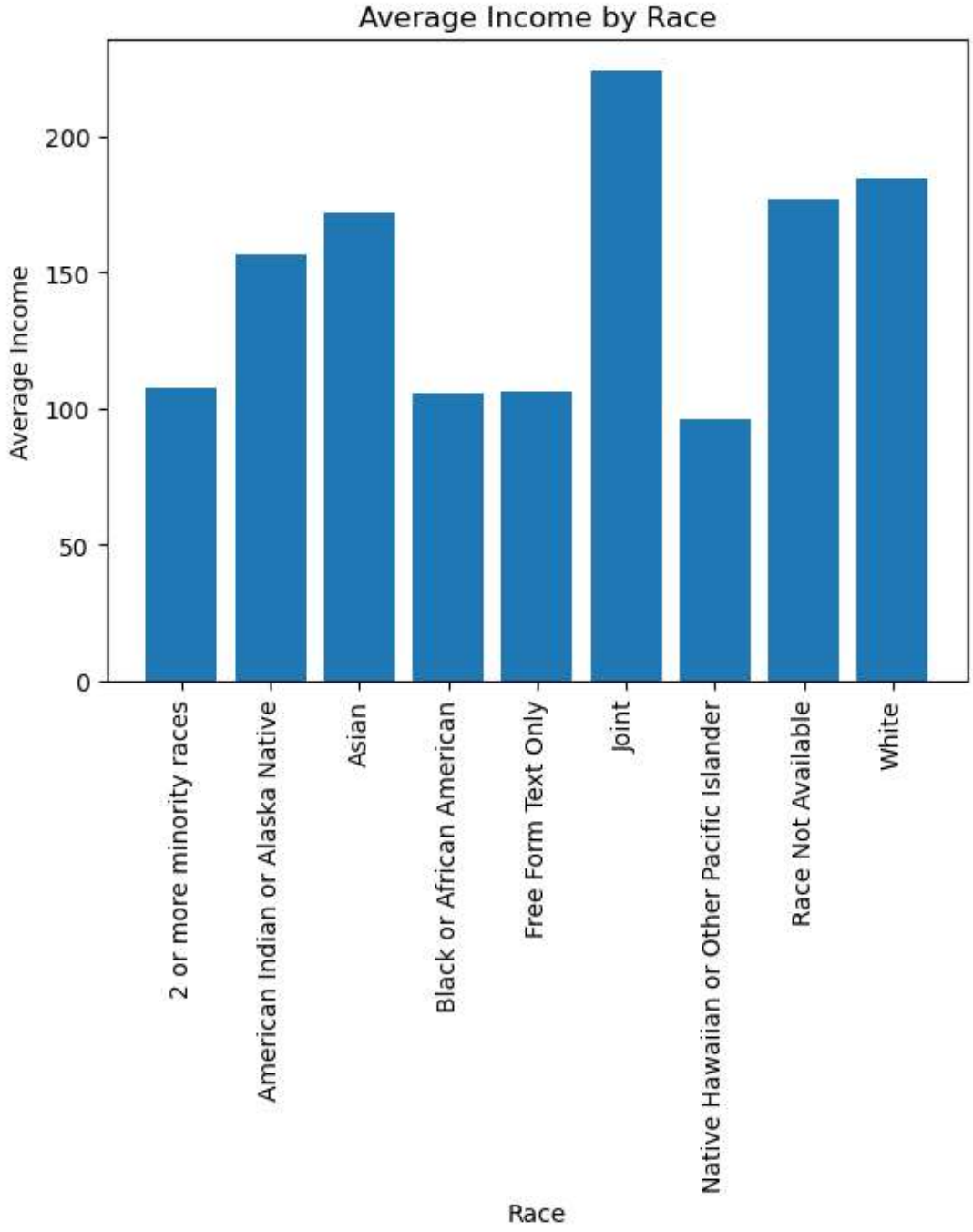
```
rejected.plot(kind='bar')
plt.xlabel('Derived Dwelling Category')
plt.ylabel('Mean Income')
plt.title('Mean Income by Derived Dwelling Category: Denied')
plt.show()
```



Average Income by Race

Entrée [21]:

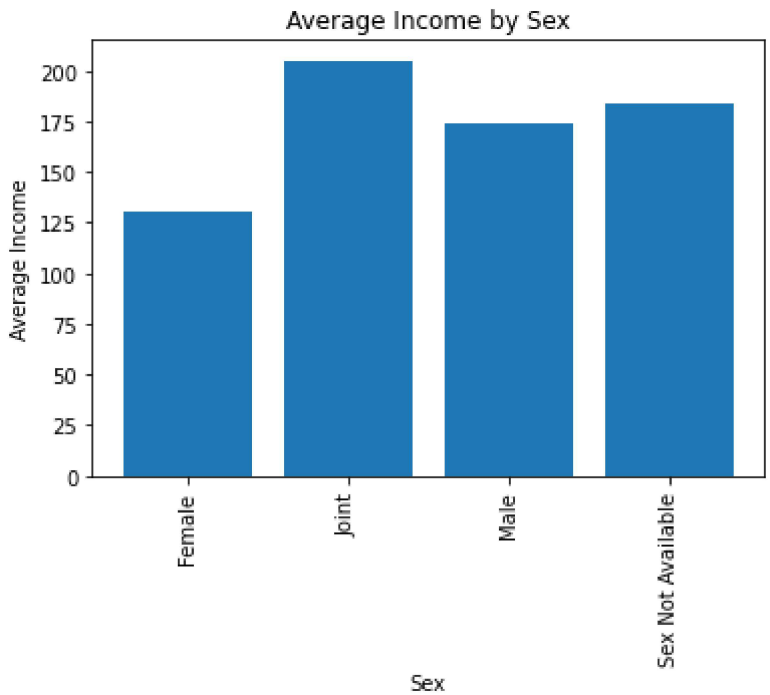
```
mean_income = df.groupby('derived_race')['income'].mean()

fig, ax = plt.subplots()
ax.bar(mean_income.index, mean_income.values)
ax.set_xlabel('Race')
ax.set_ylabel('Average Income')
ax.set_title('Average Income by Race')

plt.xticks(rotation=90)
plt.show()
```


Average Income by Race
Average Income
0
50
100
150
200
2 or more minority races
American Indian or Alaska Native
Asian
Black or African American
Free Form Text Only
Joint
Native Hawaiian or Other Pacific Islander
Race Not Available
White
Race

Average Income by Sex

Entrée [11]:

```
mean_income = df.groupby('derived_sex')['income'].mean()

fig, ax = plt.subplots()
ax.bar(mean_income.index, mean_income.values)
ax.set_xlabel('Sex')
ax.set_ylabel('Average Income')
ax.set_title('Average Income by Sex')

plt.xticks(rotation=90)
plt.show()
```

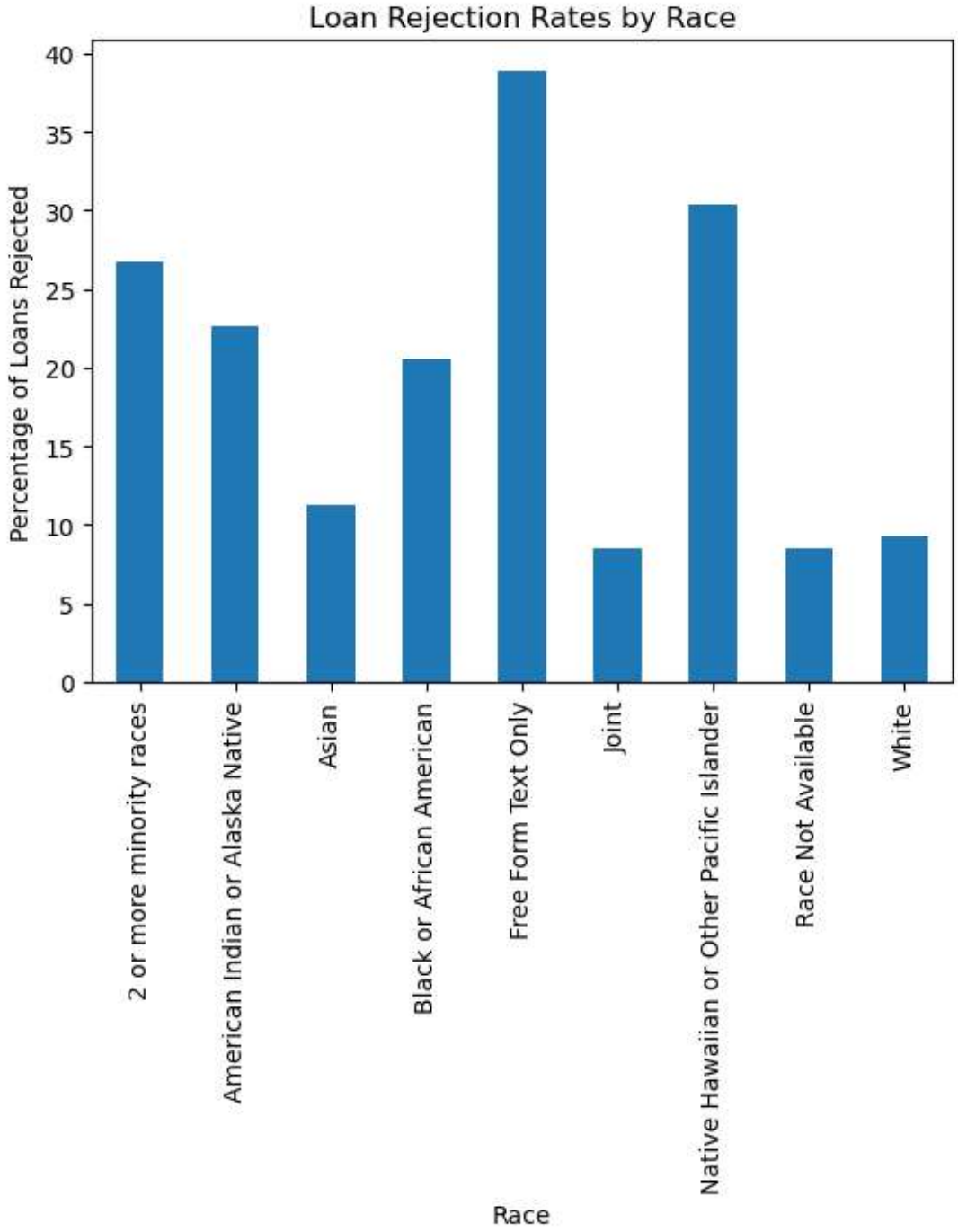


The percentage of loans that were rejected out of total loans for each race

Entrée [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
df = pd.read_csv("msamd_14454.csv")

# Calculate the percentage of rejected loans for each race
race_counts = df.groupby('derived_race')['action_taken'].value_counts(
race_rejection_rates = race_counts.loc[(slice(None), 3)] * 100

# Create a bar plot of the rejection rates by race
fig, ax = plt.subplots()
race_rejection_rates.plot(kind='bar', ax=ax)
ax.set_xlabel('Race')
ax.set_ylabel('Percentage of Loans Rejected')
ax.set_title('Loan Rejection Rates by Race')

plt.show()
```

```
C:\Users\newbee\AppData\Local\Temp\ipykernel_16280\2270341900.py:5: Dt
ypeWarning: Columns (22,23,24,26,27,28,29,30,31,32,33,38,43,44) have m
ixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv("msamd_14454.csv")
```

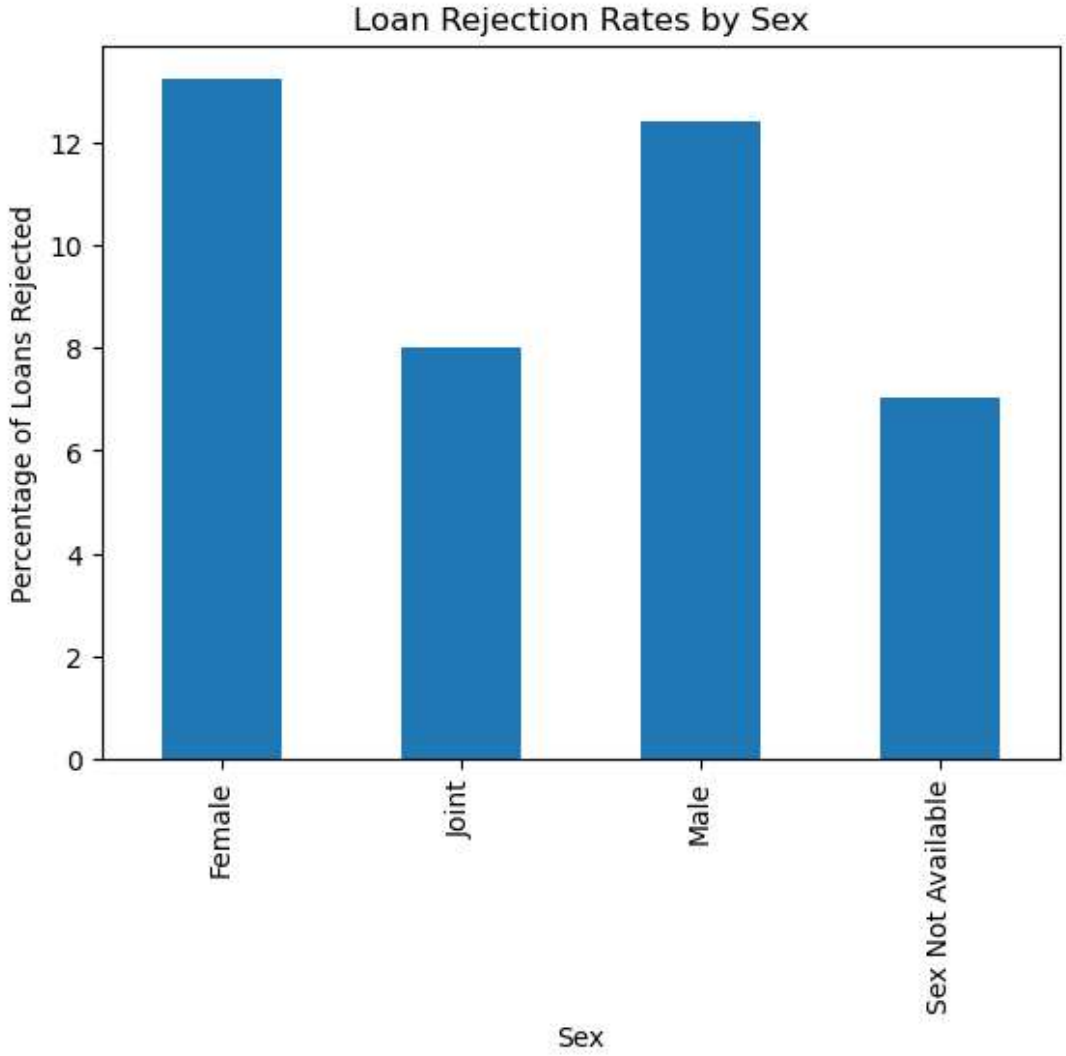

Loan Rejection Rates by Race
Percentage of Loans Rejected
0
5
10
15
20
25
30
35
40
2 or more minority races
American Indian or Alaska Native
Asian
Black or African American
Free Form Text Only
Joint
Native Hawaiian or Other Pacific Islander
Race Not Available
White
Race

The percentage of loans that were rejected out of total loans for each sex

Entrée [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
df = pd.read_csv("msamd_14454.csv")

# Calculate the percentage of rejected loans for each sex
sex_counts = df.groupby('derived_sex')['action_taken'].value_counts(no
sex_rejection_rates = sex_counts.loc[(slice(None), 3)] * 100

# Create a bar plot of the rejection rates by sex
fig, ax = plt.subplots()
sex_rejection_rates.plot(kind='bar', ax=ax)
ax.set_xlabel('Sex')
ax.set_ylabel('Percentage of Loans Rejected')
ax.set_title('Loan Rejection Rates by Sex')

plt.show()
```

```
C:\Users\newbee\AppData\Local\Temp\ipykernel_16280\1770994689.py:5: Dt
ypeWarning: Columns (22,23,24,26,27,28,29,30,31,32,33,38,43,44) have m
ixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv("msamd_14454.csv")
```


Loan Rejection Rates by Sex
Percentage of Loans Rejected
0
2
4
6
8
10
12
Female
Joint
Male
Sex Not Available
Sex

Entrée [ ]: