

## Team 4 Data Analysis

Entrée [1]:  `import pandas as pd`

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

C:\Users\perronr\AppData\Roaming\Python\Python38\site-packages\pandas\core\computation\expressions.py:20: UserWarning: Pandas requires version '2.7.3' or newer of 'numexpr' (version '2.7.1' currently installed).

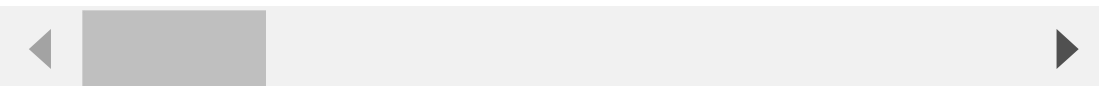
```
from pandas.core.computation.check import NUMEXPR_INSTALLED
<ipython-input-1-d37e26c08a4c>:3: DtypeWarning: Columns (22,23,24,26,27,28,29,30,31,32,33,38,43,44) have mixed types. Specify dtype option on import or set low_memory=False.
```

```
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

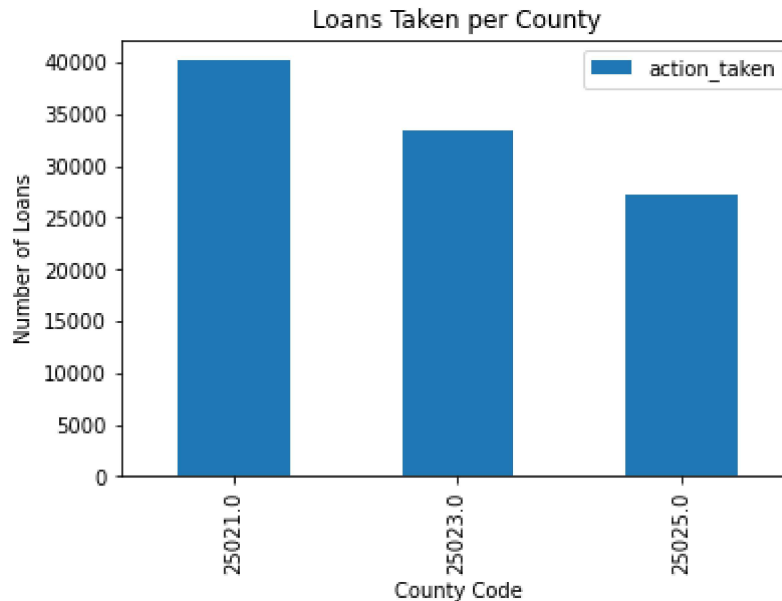
5 rows × 99 columns



### By county determine how many loans were taken.

A loan being taken has a value of 1 in the 'action\_taken' column

```
Entrée [23]: ▶ 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')
plt.title('Loans Taken per County')
plt.xlabel('County Code')
plt.ylabel('Number of Loans')
plt.show()
```

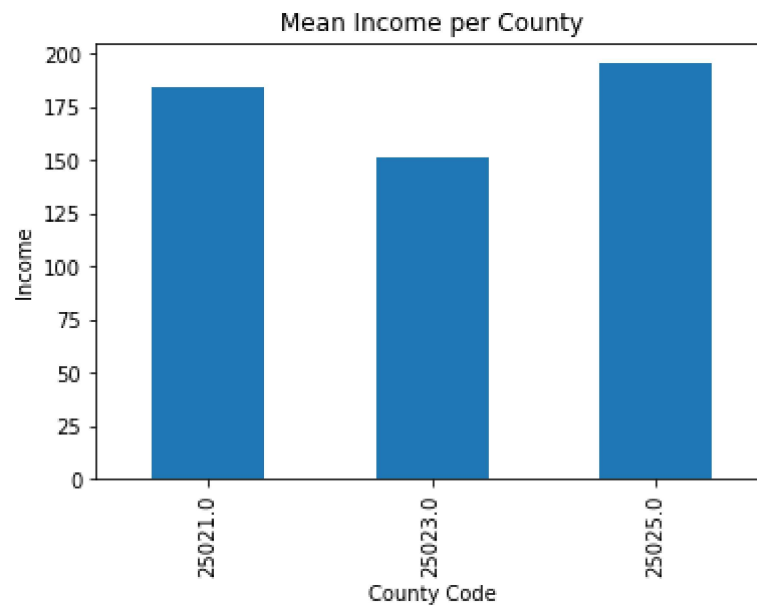


```
Entrée [3]: ▶ county_codes_dict = {25021: 'Norfolk', 25023: 'Plymouth', 25025: 'Suff
per_county_count = df3['county_code'].value_counts()

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
```

```
Entrée [20]: counties = df.groupby('county_code')['income'].mean()  
counties.plot(kind='bar')  
plt.title('Mean Income per County')  
plt.xlabel('County Code')  
plt.ylabel('Income')  
plt.show()
```



## 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_dwelling_category'] == 'Single Family (1-4 Units):Site-Built')
                (df['derived_dwelling_category'] == 'Single Family (1-4 Units):Site-Built'))]
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

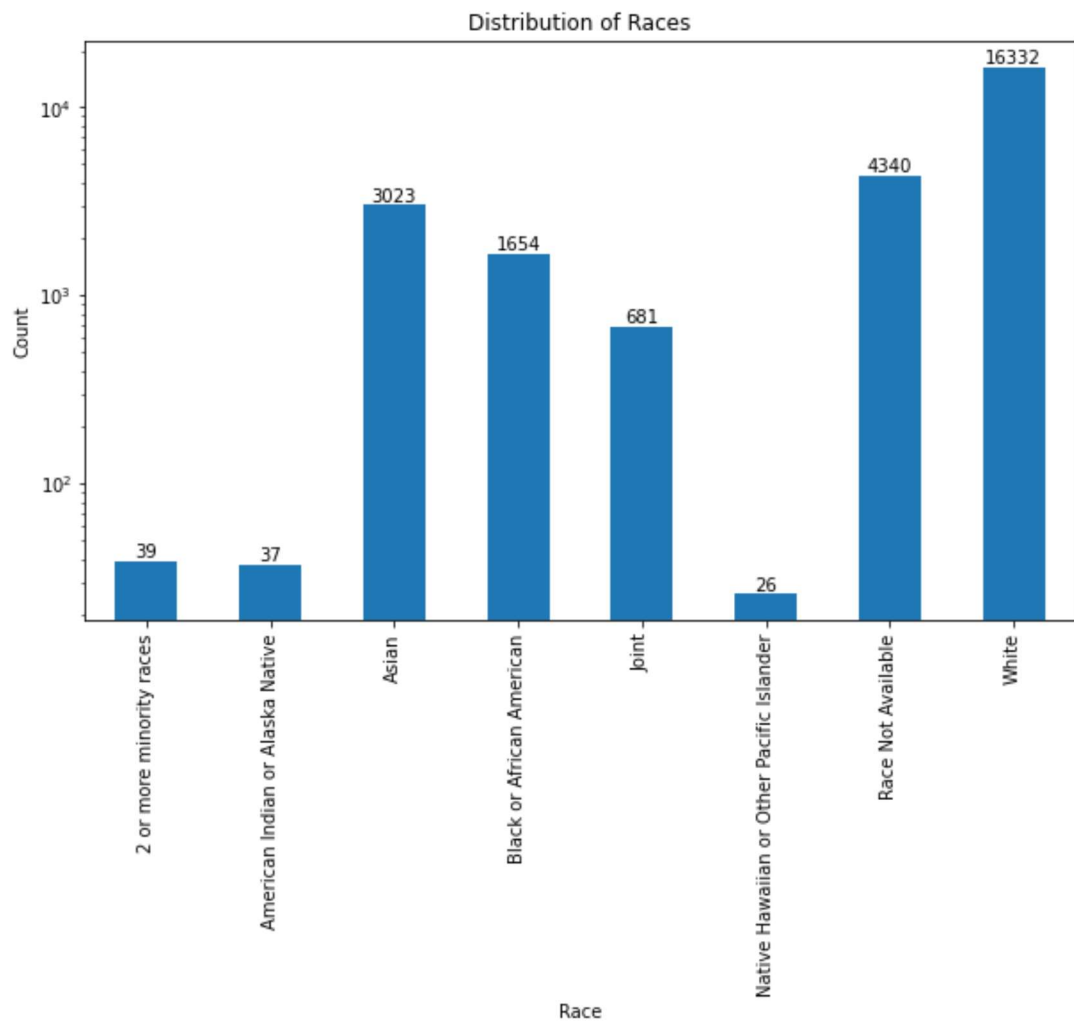
```
Entrée [22]: ▶ race = df2.groupby('derived_race')['derived_race'].count()

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

plt.yscale('log')

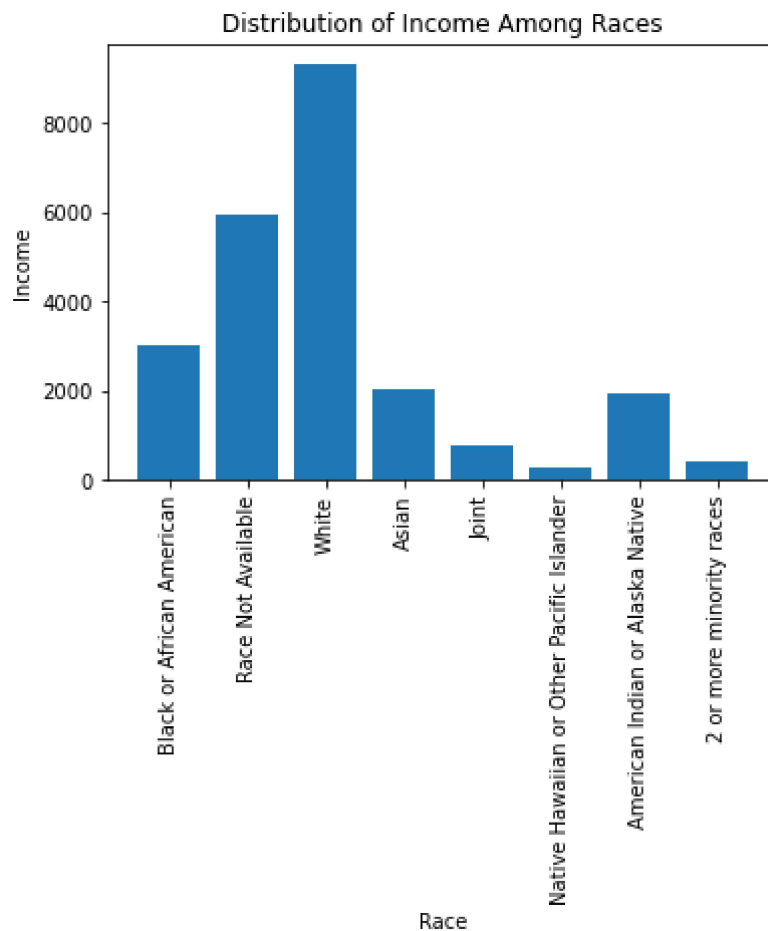
for i, v in enumerate(race):
    plt.annotate(str(v), xy=(i, v), ha='center', va='bottom')

plt.show()
```



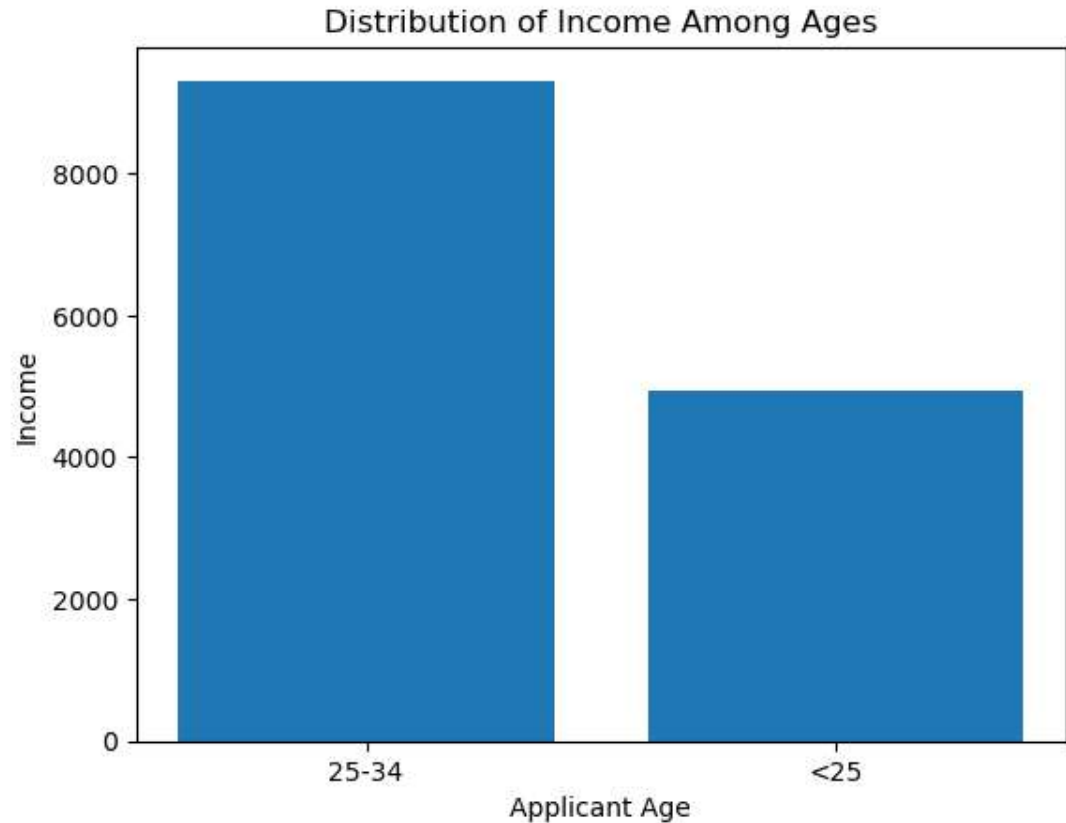
## The distribution of income for different derived race.

```
Entrée [11]: ▶ 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 [12]: ▶ new_df=df.groupby('derived_dwelling_category')
```

```
Entrée [13]: ▶ df['denial_reason-1'].unique()
```

```
Out[13]: array([ 10,  9,  3,  1,  4,  2,  5,  7,  6, 1111,
                8],
        dtype=int64)
```

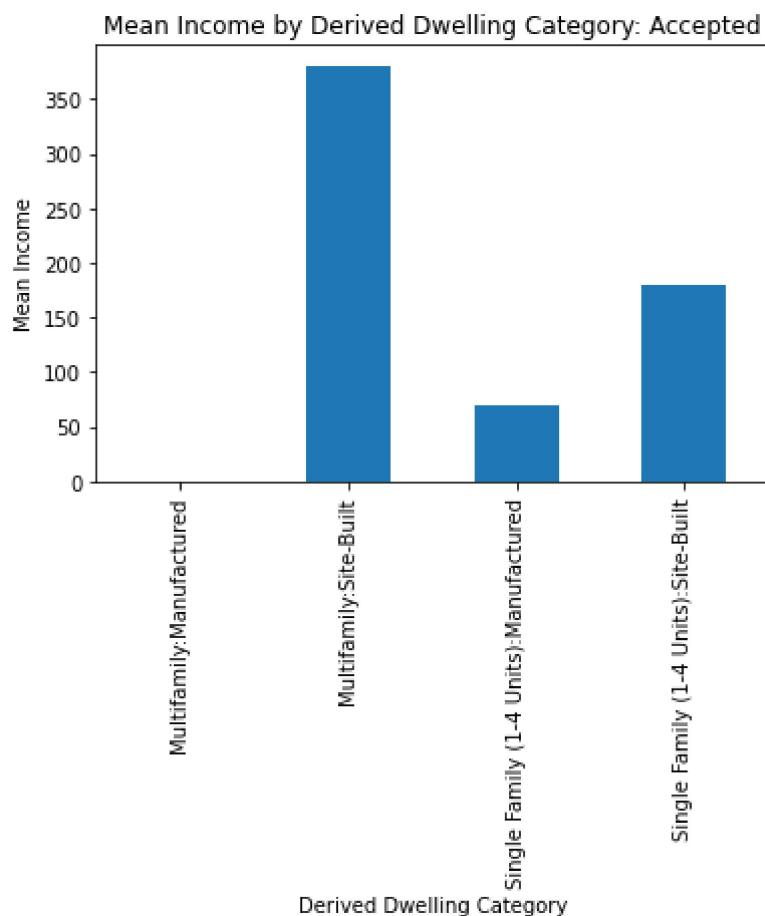
```
Entrée [14]: ▶ accepted = df.loc[df['denial_reason-1']==10]
```

```
Entrée [15]:  rejected = df.loc[df['denial_reason-1']!=10]
```

```
Entrée [16]:  accepted = accepted.groupby('derived_dwelling_category')['income'].mean()
```

```
Entrée [17]:  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()
```

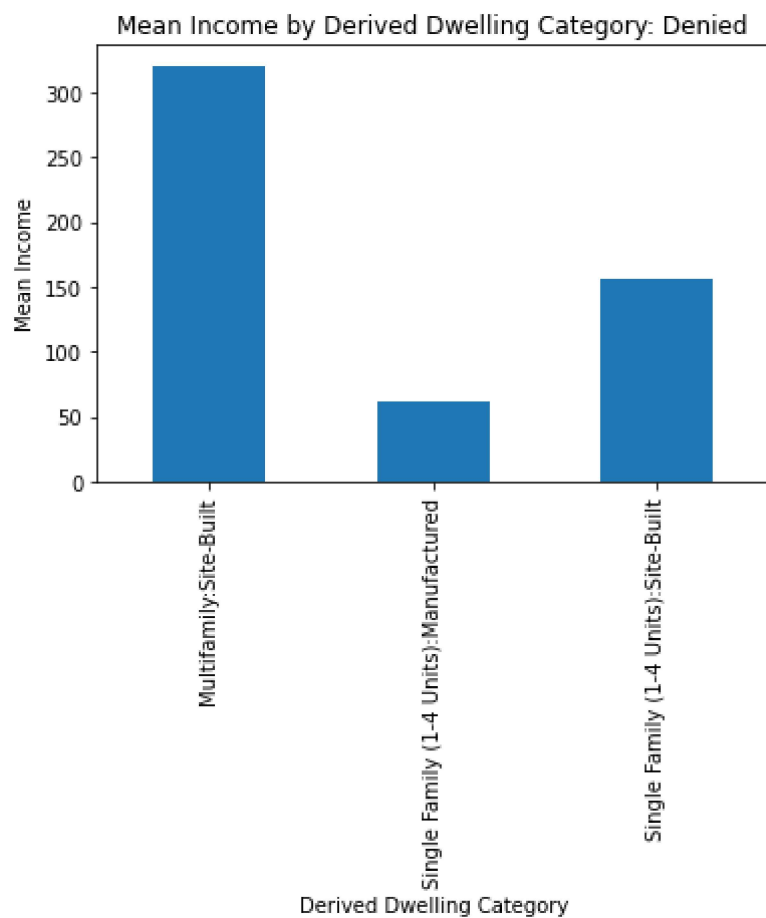


## Average Income by Housing Type: Denied

```
Entrée [18]:  rejected = rejected.groupby('derived_dwelling_category')['income'].mean()
```



```
Entrée [19]: rejected.plot(kind='bar')  
plt.xlabel('Derived Dwelling Category')  
plt.ylabel('Mean Income')  
plt.title('Mean Income by Derived Dwelling Category: Denied')  
plt.show()
```

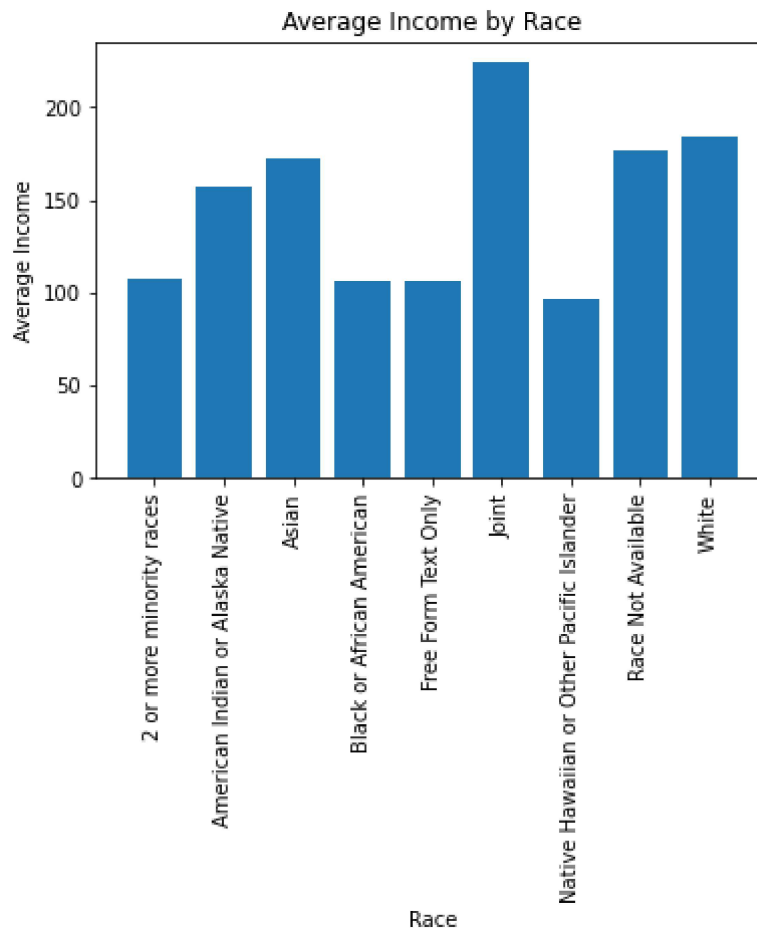


## Average Income by Race

```
Entrée [20]: 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()
```

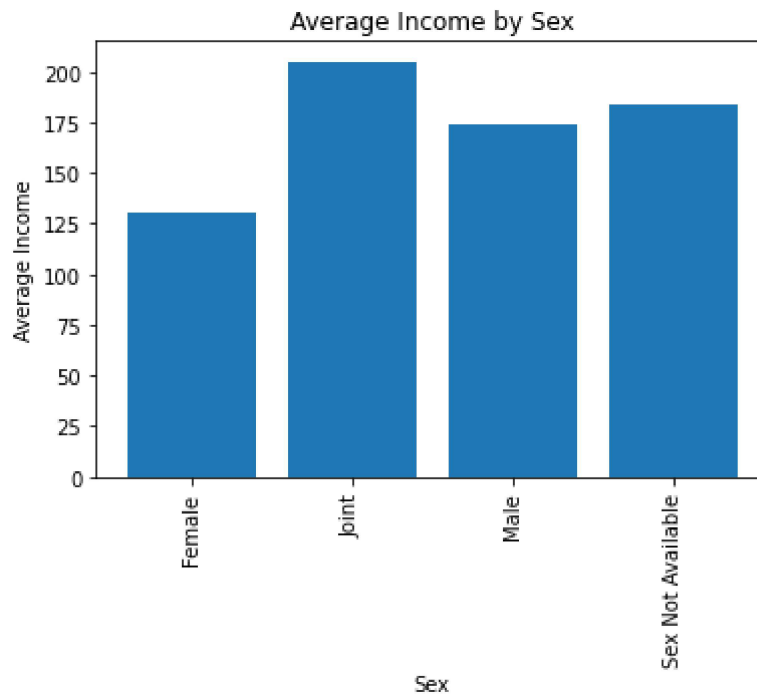


## Average Income by Sex

```
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 [21]: ▶ import pandas as pd
import matplotlib.pyplot as plt

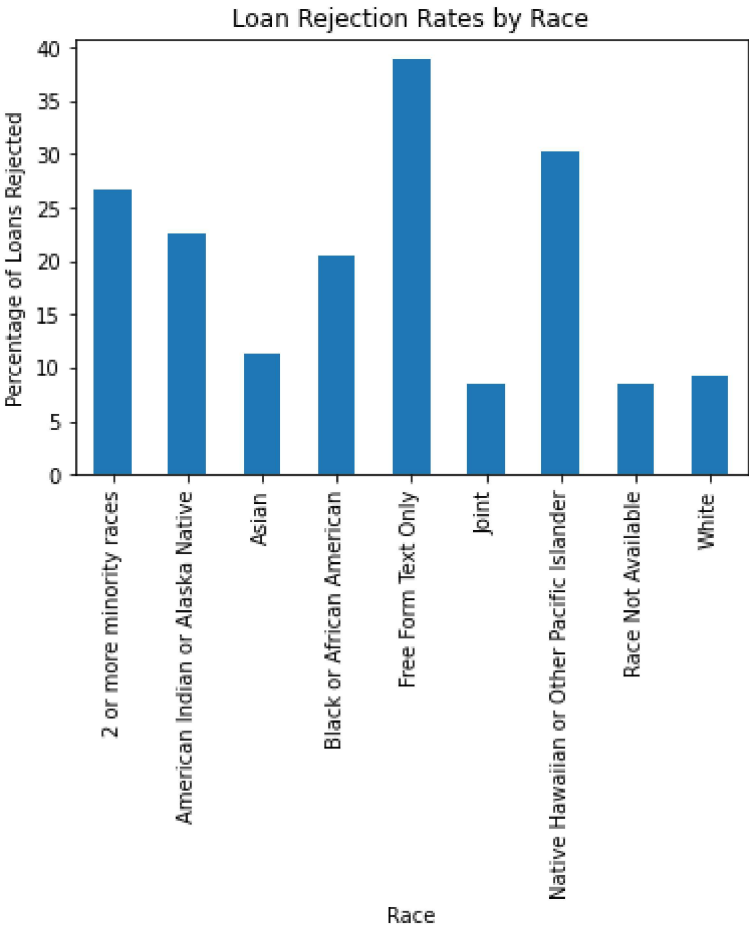
# Load the data
df = pd.read_csv("msamd_14454.csv")

# 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()
```

```
<ipython-input-21-f3771fc02029>:5: DtypeWarning: Columns (22,23,24,26,
27,28,29,30,31,32,33,38,43,44) have mixed types. Specify dtype option
on import or set low_memory=False.
    df = pd.read_csv("msamd_14454.csv")
```



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

```
Entrée [24]: ▶ import pandas as pd
import matplotlib.pyplot as plt

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

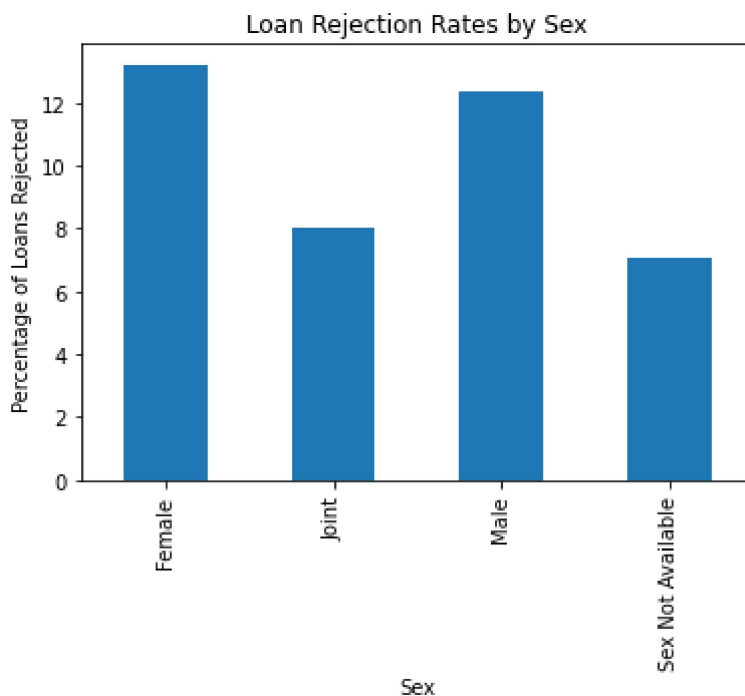
# Calculate the percentage of rejected loans for each sex
sex_counts = df.groupby('derived_sex')['action_taken'].value_counts(normalized=True)
sex_rejection_rates = sex_counts.loc[:, 3] * 100

# 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()
```

<ipython-input-24-d20213709a42>:5: DtypeWarning: Columns (22,23,24,26,27,28,29,30,31,32,33,38,43,44) have mixed types. Specify dtype option on import or set low\_memory=False.

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



## The percentage of loans that were rejected out of total loans for each loan type

Entrée [27]:

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

# Calculate the percentage of rejected Loans for each sex
type_counts = df.groupby('loan_type')['action_taken'].value_counts(normalize=True)
type_rejection_rates = type_counts.loc[(slice(None), 3)] * 100

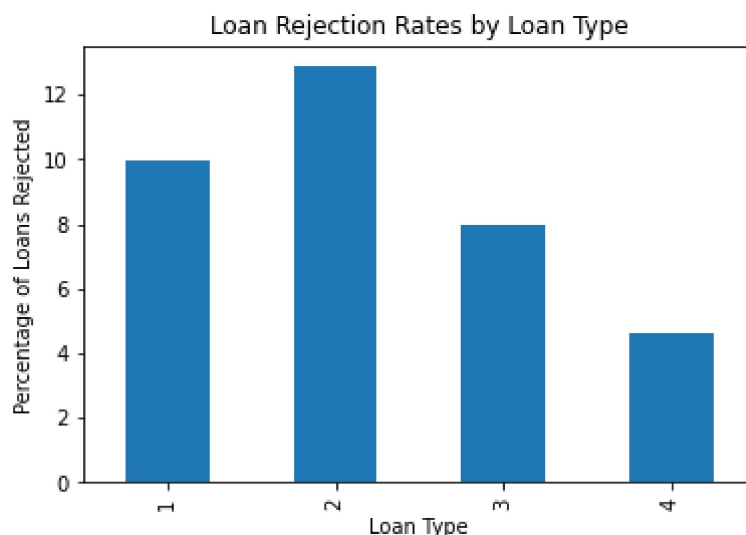
# Create a bar plot of the rejection rates by sex
fig, ax = plt.subplots()
type_rejection_rates.plot(kind='bar', ax=ax)
ax.set_xlabel('Loan Type')
ax.set_ylabel('Percentage of Loans Rejected')
ax.set_title('Loan Rejection Rates by Loan Type')

plt.show()

print("1 - Conventional (not insured or guaranteed by FHA, VA, RHS, or FSA)")
print("2 - Federal Housing Administration insured (FHA)")
print("3 - Veterans Affairs guaranteed (VA)")
print("4 - USDA Rural Housing Service or Farm Service Agency guaranteed (RHS or FSA)")
```

<ipython-input-27-3171d587b318>:2: DtypeWarning: Columns (22,23,24,26, 27,28,29,30,31,32,33,38,43,44) have mixed types. Specify dtype option on import or set low\_memory=False.

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



1 - Conventional (not insured or guaranteed by FHA, VA, RHS, or FSA)  
 2 - Federal Housing Administration insured (FHA)  
 3 - Veterans Affairs guaranteed (VA)  
 4 - USDA Rural Housing Service or Farm Service Agency guaranteed (RHS or FSA)

## Most Common Reasons for Loan Rejection

```
Entrée [17]: ▶ # Load the data
import pandas as pd
from matplotlib import pyplot as plt

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

# Calculate the percentage of rejected Loans for each sex
denial = df.groupby('denial_reason-1')['denial_reason-1'].count()

denial = denial[:9]

# Create a bar plot of the rejection rates by sex
fig, ax = plt.subplots()
denial.plot(kind='bar', ax=ax)
ax.set_xlabel('Denial Reason')
ax.set_ylabel('Number of Loans Rejected')
ax.set_title('Number of Loans Rejected for Each Denial Reason')

plt.show()
```

<ipython-input-17-004e07747232>:5: DtypeWarning: Columns (22,23,24,26, 27,28,29,30,31,32,33,38,43,44) have mixed types. Specify dtype option on import or set low\_memory=False.

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

