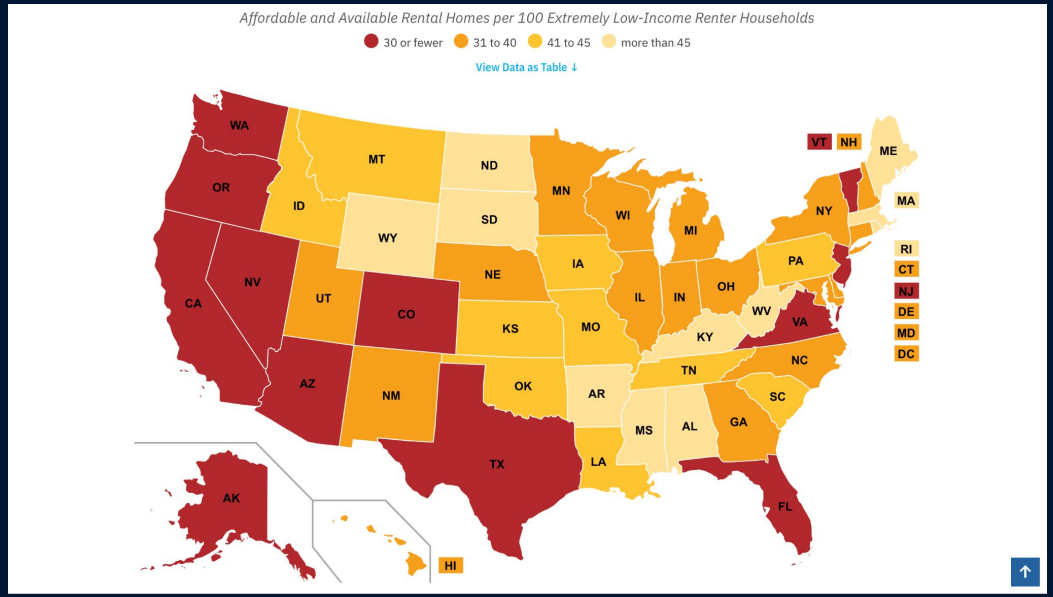
Income vs. Housing

Justin Zavala
CTEC 426

Problem

Housing affordability remains a critical issue for both renters and homeowners, with many households experiencing housing cost burdens, spending more than 30% of their income on rent or mortgage costs.

National Low Income Housing Coalition (NLIHC)



Questions

1. What percent of households are cost-burdened?
2. How does housing cost burden vary across income levels?
3. Are renters or homeowners more likely to experience severe housing cost burden?
4. How does housing cost burden vary by urban vs. rural areas?
5. Are specific demographics more likely to experience severe housing cost burden?

Variables

1. HINCP (Household Income)
2. ADJINC (Inflation-Adjusted Income)
3. GRPIP – (Gross Rent as a Percentage of Household Income)
4. RAC1P – Race of the householder
5. AGEP – Age of the householder (used to categorize the age groups).
6. YRBLT – Year structure built.
7. NP - Household size
8. ESR Employment Status
9. MRGP - Mortgage Payment
10. RNTP - Monthly Rent

Ingestion

- Data is extracted from the American Community Survey (ACS) 2023 1-year dataset
- Two separate files were used: Household file (husa.csv) and Person file (pusa.csv)
- Both Household and Person file had 287 features

Wrangling

- Data Cleaning
- Data Integration
- Data Reduction
- Data Validation
- Feature Engineering

Literature Review

Housing Cost Burden Across Income Levels

Definition: Spending >30% of income on housing; severe burden >50%.

Key Disparities:

- Income
- Race
- Housing Age

- Market & Policy Factor
- Research Gaps.

Review Questions

- Housing Cost Burden & Income Disparities
- Racial & Demographic Disparities
- Housing Age & Structural Factors
- Policy & Market Influences
- Mortgage & Homeownership Trends

Let's Explore

- Variables include income, housing costs, & demographic and household.
- The final analytic dataset will have roughly eight to ten features
- Feature Engineering
  - 0 = No burden (<30% of income)
  - 1 = Burden (30% - 49% of income
  - 2 = Severe burden (>50% of income)
- Missing data, outliers, and Inconsistencies

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#load the data
df = pd.read_csv('//Users//justinzavala//Documents//Justin Dataset 2//psam_pusa.csv')

#print the first 5 rows of the dataframe
print(df.head(20))
```

```
   RT       SERIALNO  DIVISION  SPORDER  PUMA  REGION  STATE    ADJINC  PWGTP  \
0   P  2023GQ0000113         6        1  2802       3      1  1019518      6
1   P  2023GQ0000180         6        1   100       3      1  1019518     27
2   P  2023GQ0000181         6        1   402       3      1  1019518     47
3   P  2023GQ0000250         6        1  2702       3      1  1019518     11
4   P  2023GQ0000340         6        1  2802       3      1  1019518     57
5   P  2023GQ0000364         6        1  1202       3      1  1019518     52
6   P  2023GQ0000493         6        1  1801       3      1  1019518     43
7   P  2023GQ0000537         6        1  1402       3      1  1019518      1
8   P  2023GQ0000558         6        1  1404       3      1  1019518      3
9   P  2023GQ0000796         6        1  2802       3      1  1019518     41
10  P  2023GQ0000857         6        1  2803       3      1  1019518     43
11  P  2023GQ0001001         6        1  2101       3      1  1019518     39
12  P  2023GQ0001251         6        1   800       3      1  1019518     36
13  P  2023GQ0001264         6        1  1801       3      1  1019518     65
14  P  2023GQ0001265         6        1  2600       3      1  1019518     47
15  P  2023GQ0001272         6        1  1901       3      1  1019518     45
16  P  2023GQ0001398         6        1  1801       3      1  1019518     45
17  P  2023GQ0001448         6        1  1100       3      1  1019518     77
18  P  2023GQ0001451         6        1  1502       3      1  1019518     48
19  P  2023GQ0001468         6        1  1404       3      1  1019518     26

    AGEP  ...  PWGTP71  PWGTP72  PWGTP73  PWGTP74  PWGTP75  PWGTP76  PWGTP77  \
0     86  ...        7        7        5        5        8        7        7
1     60  ...       25       18       23       28       30       22       25
2     20  ...       42        1       57        1       59       58       38
3     13  ...       11        9       11       11       11        9       12
4     18  ...       61       54       59        1       50        1       55
5     19  ...       53       91       50        0      103       62       98
6     37  ...       46       40       36       42       60       73       62
7     71  ...        1        1        2        1        1        1        1
8     75  ...        2        2        2        2        2        2        2
9     19  ...        0       31        1       63       38       40       64
10    37  ...       60       48       56       72       50       87        6
11    18  ...       37       31       39       45       44       45       39
12    59  ...       34        4       26        7       44       35        7
13    72  ...       64       64       65       66       65       66       67
14    33  ...       84       36        6       94       88       53       47
15    49  ...       37       73       49       39       79       50        7
16    18  ...       68        0       37        0       41        1        0
17    45  ...      150      124       81       74       75       81       19
18    36  ...       87       75        9       47       11       32       48
19    44  ...       51        3       25       26       50       25       50

    PWGTP78  PWGTP79  PWGTP80
0         6        6        7
1        29       23       18
2        52        1       83
3        16       17       13
4        70        1        0
5        52        0       92
6        16       32       11
7         0        0        1
8         2        3        2
9        50       77       82
10        6       33       39
11       38       35       40
```

[4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#load the (HUSA)data
df = pd.read_csv('//Users//justinzavala//Documents//Justin Dataset 2//psam_husa.csv')

#print the first 5 rows of the dataframe
print(df.head(20))
```

```
   RT       SERIALNO  DIVISION  PUMA  REGION  STATE   ADJHSG   ADJINC  WGTP  \
0   H  2023GQ0000113         6  2802       3      1  1000000  1019518     0
1   H  2023GQ0000180         6   100       3      1  1000000  1019518     0
2   H  2023GQ0000181         6   402       3      1  1000000  1019518     0
3   H  2023GQ0000250         6  2702       3      1  1000000  1019518     0
4   H  2023GQ0000340         6  2802       3      1  1000000  1019518     0
5   H  2023GQ0000364         6  1202       3      1  1000000  1019518     0
6   H  2023GQ0000493         6  1801       3      1  1000000  1019518     0
7   H  2023GQ0000537         6  1402       3      1  1000000  1019518     0
8   H  2023GQ0000558         6  1404       3      1  1000000  1019518     0
9   H  2023GQ0000796         6  2802       3      1  1000000  1019518     0
10  H  2023GQ0000857         6  2803       3      1  1000000  1019518     0
11  H  2023GQ0001001         6  2101       3      1  1000000  1019518     0
12  H  2023GQ0001251         6   800       3      1  1000000  1019518     0
13  H  2023GQ0001264         6  1801       3      1  1000000  1019518     0
14  H  2023GQ0001265         6  2600       3      1  1000000  1019518     0
15  H  2023GQ0001272         6  1901       3      1  1000000  1019518     0
16  H  2023GQ0001398         6  1801       3      1  1000000  1019518     0
17  H  2023GQ0001448         6  1100       3      1  1000000  1019518     0
18  H  2023GQ0001451         6  1502       3      1  1000000  1019518     0
19  H  2023GQ0001468         6  1404       3      1  1000000  1019518     0

    NP  ...  WGTP71  WGTP72  WGTP73  WGTP74  WGTP75  WGTP76  WGTP77  WGTP78  \
0    1  ...       0       0       0       0       0       0       0       0
1    1  ...       0       0       0       0       0       0       0       0
2    1  ...       0       0       0       0       0       0       0       0
3    1  ...       0       0       0       0       0       0       0       0
4    1  ...       0       0       0       0       0       0       0       0
5    1  ...       0       0       0       0       0       0       0       0
6    1  ...       0       0       0       0       0       0       0       0
7    1  ...       0       0       0       0       0       0       0       0
8    1  ...       0       0       0       0       0       0       0       0
9    1  ...       0       0       0       0       0       0       0       0
10   1  ...       0       0       0       0       0       0       0       0
11   1  ...       0       0       0       0       0       0       0       0
12   1  ...       0       0       0       0       0       0       0       0
13   1  ...       0       0       0       0       0       0       0       0
14   1  ...       0       0       0       0       0       0       0       0
15   1  ...       0       0       0       0       0       0       0       0
16   1  ...       0       0       0       0       0       0       0       0
17   1  ...       0       0       0       0       0       0       0       0
18   1  ...       0       0       0       0       0       0       0       0
19   1  ...       0       0       0       0       0       0       0       0

    WGTP79  WGTP80
0        0       0
1        0       0
2        0       0
3        0       0
4        0       0
5        0       0
6        0       0
7        0       0
8        0       0
9        0       0
10       0       0
11       0       0
12       0       0
```

```
10      0       0
11      0       0
12      0       0
13      0       0
14      0       0
15      0       0
16      0       0
17      0       0
18      0       0
19      0       0

[20 rows x 241 columns]
```

[8]:
```
import pandas as pf

pusa = pd.read_csv('//Users//justinzavala//Documents//Justin Dataset 2//psam_pusa.csv')

print(pusa.head(5))
```

```
  RT       SERIALNO  DIVISION  SPORDER  PUMA  REGION  STATE   ADJINC  PWGTP  \
0  P  2023GQ0000113         6        1  2802       3      1  1019518      6
1  P  2023GQ0000180         6        1   100       3      1  1019518     27
2  P  2023GQ0000181         6        1   402       3      1  1019518     47
3  P  2023GQ0000250         6        1  2702       3      1  1019518     11
4  P  2023GQ0000340         6        1  2802       3      1  1019518     57

   AGEP  ...  PWGTP71  PWGTP72  PWGTP73  PWGTP74  PWGTP75  PWGTP76  PWGTP77  \
0    86  ...        7        7        5        5        8        7        7
1    60  ...       25       18       23       28       30       22       25
2    20  ...       42        1       57        1       59       58       38
3    13  ...       11        9       11       11       11        9       12
4    18  ...       61       54       59        1       50        1       55

   PWGTP78  PWGTP79  PWGTP80
0        6        6        7
1       29       23       18
2       52        1       83
3       16       17       13
4       70        1        0

[5 rows x 287 columns]
```

[12…
```
import pandas as pd

husa = pd.read_csv('//Users//justinzavala//Documents//Justin Dataset 2//psam_husa.csv')

print(husa.head(5))
```

```
  RT       SERIALNO  DIVISION  PUMA  REGION  STATE   ADJHSG   ADJINC  WGTP  \
0  H  2023GQ0000113         6  2802       3      1  1000000  1019518     0
1  H  2023GQ0000180         6   100       3      1  1000000  1019518     0
2  H  2023GQ0000181         6   402       3      1  1000000  1019518     0
3  H  2023GQ0000250         6  2702       3      1  1000000  1019518     0
4  H  2023GQ0000340         6  2802       3      1  1000000  1019518     0

   NP  ...  WGTP71  WGTP72  WGTP73  WGTP74  WGTP75  WGTP76  WGTP77  WGTP78  \
0   1  ...       0       0       0       0       0       0       0       0
1   1  ...       0       0       0       0       0       0       0       0
2   1  ...       0       0       0       0       0       0       0       0
3   1  ...       0       0       0       0       0       0       0       0
4   1  ...       0       0       0       0       0       0       0       0

   WGTP79  WGTP80
0       0       0
1       0       0
2       0       0
3       0       0
4       0       0

[5 rows x 241 columns]
```

[13…
```
#Merge the PUSA and HUSA File on SERIALNO
merged_df = pd.merge(pusa, husa, on="SERIALNO", how="left") #One to Many Merge

#Display First 5 few rows
print(merged_df.head(20))
```

```
   RT_x       SERIALNO  DIVISION_x  SPORDER  PUMA_x  REGION_x  STATE_x  \
0     P  2023GQ0000113           6        1    2802         3        1
1     P  2023GQ0000180           6        1     100         3        1
2     P  2023GQ0000181           6        1     402         3        1
3     P  2023GQ0000250           6        1    2702         3        1
4     P  2023GQ0000340           6        1    2802         3        1
5     P  2023GQ0000364           6        1    1202         3        1
6     P  2023GQ0000493           6        1    1801         3        1
7     P  2023GQ0000537           6        1    1402         3        1
8     P  2023GQ0000558           6        1    1404         3        1
9     P  2023GQ0000796           6        1    2802         3        1
10    P  2023GQ0000857           6        1    2803         3        1
11    P  2023GQ0001001           6        1    2101         3        1
12    P  2023GQ0001251           6        1     800         3        1
13    P  2023GQ0001264           6        1    1801         3        1
14    P  2023GQ0001265           6        1    2600         3        1
15    P  2023GQ0001272           6        1    1901         3        1
16    P  2023GQ0001398           6        1    1801         3        1
17    P  2023GQ0001448           6        1    1100         3        1
18    P  2023GQ0001451           6        1    1502         3        1
19    P  2023GQ0001468           6        1    1404         3        1

    ADJINC_x  PWGTP  AGEP  ...  WGTP71  WGTP72  WGTP73  WGTP74  WGTP75  \
0    1019518      6    86  ...       0       0       0       0       0
1    1019518     27    60  ...       0       0       0       0       0
2    1019518     47    20  ...       0       0       0       0       0
3    1019518     11    13  ...       0       0       0       0       0
4    1019518     57    18  ...       0       0       0       0       0
5    1019518     52    19  ...       0       0       0       0       0
6    1019518     43    37  ...       0       0       0       0       0
7    1019518      1    71  ...       0       0       0       0       0
8    1019518      3    75  ...       0       0       0       0       0
9    1019518     41    19  ...       0       0       0       0       0
10   1019518     43    37  ...       0       0       0       0       0
11   1019518     39    18  ...       0       0       0       0       0
12   1019518     36    59  ...       0       0       0       0       0
13   1019518     65    72  ...       0       0       0       0       0
14   1019518     47    33  ...       0       0       0       0       0
15   1019518     45    49  ...       0       0       0       0       0
16   1019518     45    18  ...       0       0       0       0       0
17   1019518     77    45  ...       0       0       0       0       0
18   1019518     48    36  ...       0       0       0       0       0
19   1019518     26    44  ...       0       0       0       0       0

    WGTP76  WGTP77  WGTP78  WGTP79  WGTP80
0        0       0       0       0       0
1        0       0       0       0       0
2        0       0       0       0       0
3        0       0       0       0       0
4        0       0       0       0       0
5        0       0       0       0       0
6        0       0       0       0       0
7        0       0       0       0       0
8        0       0       0       0       0
9        0       0       0       0       0
10       0       0       0       0       0
11       0       0       0       0       0
12       0       0       0       0       0
13       0       0       0       0       0
14       0       0       0       0       0
15       0       0       0       0       0
16       0       0       0       0       0
17       0       0       0       0       0
18       0       0       0       0       0
```

[20 rows x 527 columns]

[15…]
```
#Check the column names in the merged dataset
print(merged_df.columns)
```

```
Index(['RT_x', 'SERIALNO', 'DIVISION_x', 'SPORDER', 'PUMA_x', 'REGION_x',
       'STATE_x', 'ADJINC_x', 'PWGTP', 'AGEP',
       ...
       'WGTP71', 'WGTP72', 'WGTP73', 'WGTP74', 'WGTP75', 'WGTP76', 'WGTP77',
       'WGTP78', 'WGTP79', 'WGTP80'],
      dtype='object', length=527)
```

[18…]
```
# List of columns you want to keep
columns_to_keep = [
    'HINCP',  # Household Income
    'GRPIP',  # Gross Income
    'AGEP',  # Age
    'YRBLT',  # Year Built
    'NP',  # Number of People
    'ESR',  # Employment Status
    'RAC1P',  # Race
    'MRGP',  # Mortgage
    'RNTP'  # Rent
]

# Select only the relevant columns
selected_data = merged_df[columns_to_keep]

# Check the first few rows to confirm
print(selected_data.head(20))
```

	HINCP	GRPIP	AGEP	YRBLT	NP	ESR	RAC1P	MRGP	RNTP
0	NaN	NaN	86	NaN	1	6.0	2	NaN	NaN
1	NaN	NaN	60	NaN	1	6.0	1	NaN	NaN
2	NaN	NaN	20	NaN	1	6.0	1	NaN	NaN
3	NaN	NaN	13	NaN	1	NaN	2	NaN	NaN
4	NaN	NaN	18	NaN	1	6.0	1	NaN	NaN
5	NaN	NaN	19	NaN	1	6.0	1	NaN	NaN
6	NaN	NaN	37	NaN	1	6.0	2	NaN	NaN
7	NaN	NaN	71	NaN	1	6.0	1	NaN	NaN
8	NaN	NaN	75	NaN	1	6.0	1	NaN	NaN
9	NaN	NaN	19	NaN	1	1.0	1	NaN	NaN
10	NaN	NaN	37	NaN	1	6.0	2	NaN	NaN
11	NaN	NaN	18	NaN	1	3.0	1	NaN	NaN
12	NaN	NaN	59	NaN	1	6.0	2	NaN	NaN
13	NaN	NaN	72	NaN	1	6.0	1	NaN	NaN
14	NaN	NaN	33	NaN	1	6.0	1	NaN	NaN
15	NaN	NaN	49	NaN	1	6.0	2	NaN	NaN
16	NaN	NaN	18	NaN	1	6.0	2	NaN	NaN
17	NaN	NaN	45	NaN	1	6.0	1	NaN	NaN
18	NaN	NaN	36	NaN	1	6.0	2	NaN	NaN
19	NaN	NaN	44	NaN	1	6.0	1	NaN	NaN

[ ]:

11	NaN	NaN	18	NaN	1	3.0	1	NaN	NaN
12	NaN	NaN	59	NaN	1	6.0	2	NaN	NaN
13	NaN	NaN	72	NaN	1	6.0	1	NaN	NaN
14	NaN	NaN	33	NaN	1	6.0	1	NaN	NaN
15	NaN	NaN	49	NaN	1	6.0	2	NaN	NaN
16	NaN	NaN	18	NaN	1	6.0	2	NaN	NaN
17	NaN	NaN	45	NaN	1	6.0	1	NaN	NaN
18	NaN	NaN	36	NaN	1	6.0	2	NaN	NaN
19	NaN	NaN	44	NaN	1	6.0	1	NaN	NaN

[22…]
```
# Save the selected data to a new CSV file
selected_data.to_csv("HousingVsIncome.csv", index=False)
```

[30…]
```
import pandas as pd

# Read the file you already filtered down to the 11 variables
df = pd.read_csv('//Users//justinzavala//Documents//Justin Dataset 2//HousingVsIncome.csv')

# Handling Missing Data - Column by Column Strategy

# Income and rent/mortgage amounts - Fill with median (less sensitive to outliers)
df['HINCP'].fillna(df['HINCP'].median(), inplace=True)
df['GRPIP'].fillna(df['GRPIP'].median(), inplace=True)
df['RNTP'].fillna(0, inplace=True)  # Set rent to 0 for missing (likely owners)
df['MRGP'].fillna(0, inplace=True)  # Set mortgage to 0 for missing (likely renters)

# Categorical variables - Fill with most common value (mode)
df['RAC1P'].fillna(df['RAC1P'].mode()[0], inplace=True)
df['AGEP'].fillna(df['AGEP'].mode()[0], inplace=True)
df['ESR'].fillna(df['ESR'].mode()[0], inplace=True)

# Year built - Could fill with median (reasonable assumption for age of homes)
df['YRBLT'].fillna(df['YRBLT'].median(), inplace=True)

# Household size - Fill with median (smaller households more common)
df['NP'].fillna(df['NP'].median(), inplace=True)

# Optional - Save the cleaned version if you want
df.to_csv('HousingVsIncomeCleaned.csv', index=False)

print('Missing data handled! Cleaned file saved as HousingVsCleaned.csv')
print(df.info())  # Check for remaining missing values
```

```
Missing data handled! Cleaned file saved as HousingVsCleaned.csv
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1732343 entries, 0 to 1732342
Data columns (total 9 columns):
 #   Column  Dtype
---  ------  -----
 0   HINCP   float64
 1   GRPIP   float64
 2   AGEP    int64
 3   YRBLT   float64
 4   NP      int64
 5   ESR     float64
 6   RAC1P   int64
 7   MRGP    float64
 8   RNTP    float64
dtypes: float64(6), int64(3)
memory usage: 119.0 MB
None
```

jupyter Dataset Last Checkpoint: 19 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
df['YRBLT'].fillna(df['YRBLT'].median(), inplace=True)

# Household size - Fill with median (smaller households more common)
df['NP'].fillna(df['NP'].median(), inplace=True)

# Optional - Save the cleaned version if you want
df.to_csv('HousingVsIncomeCleaned.csv', index=False)

print('Missing data handled! Cleaned file saved as acs_cleaned_variables1.csv')
print(df.info())  # Check for remaining missing values
```

```
Missing data handled! Cleaned file saved as acs_cleaned_variables1.csv
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1732343 entries, 0 to 1732342
Data columns (total 9 columns):
 #   Column  Dtype
---  ------  -----
 0   HINCP   float64
 1   GRPIP   float64
 2   AGEP    int64
 3   YRBLT   float64
 4   NP      int64
 5   ESR     float64
 6   RAC1P   int64
 7   MRGP    float64
 8   RNTP    float64
dtypes: float64(6), int64(3)
memory usage: 119.0 MB
None
```

[25…

```
import pandas as pd

# Load the cleaned dataset
df = pd.read_csv('HousingVsIncomeCleaned.csv')

# List of your selected variables
selected_columns = [
    'HINCP', 'GRPIP', 'RAC1P', 'AGEP',
    'ESR', 'YRBLT', 'NP', 'RNTP', 'MRGP'
]

# Create a new DataFrame with only those variables
df_selected = df[selected_columns]

# Save the new dataset
df_selected.to_csv('HousingVsIncomeCleaned2.csv', index=False)

print('New file saved as HousingVsIncomeCleaned2.csv with only selected variables.')
print(df_selected.head())  # Preview first few rows
```

```
New file saved as HousingVsIncomeCleaned2.csv with only selected variables.
     HINCP  GRPIP  RAC1P  AGEP  ESR   YRBLT  NP  RNTP  MRGP
0  97000.0   29.0      2    86  6.0  1980.0   1   0.0   0.0
1  97000.0   29.0      1    60  6.0  1980.0   1   0.0   0.0
2  97000.0   29.0      1    20  6.0  1980.0   1   0.0   0.0
3  97000.0   29.0      2    13  1.0  1980.0   1   0.0   0.0
4  97000.0   29.0      1    18  6.0  1980.0   1   0.0   0.0
```

[ ]:

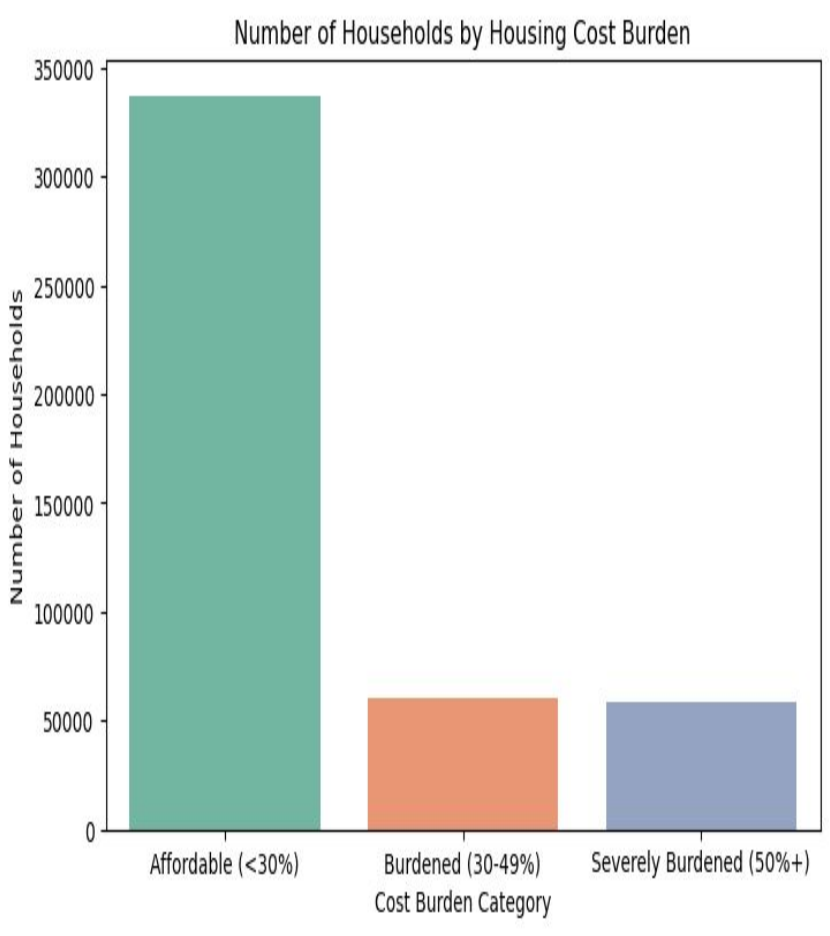
jupyter Data Visualizations Last Checkpoint: 6 days ago

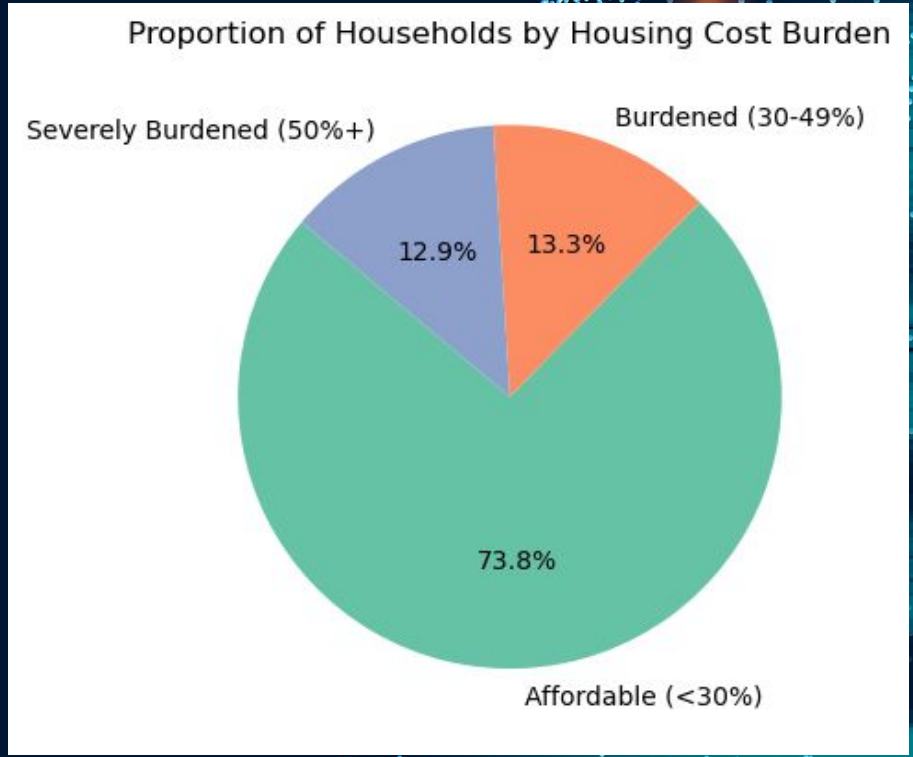
File Edit View Run Kernel Settings Help Trusted

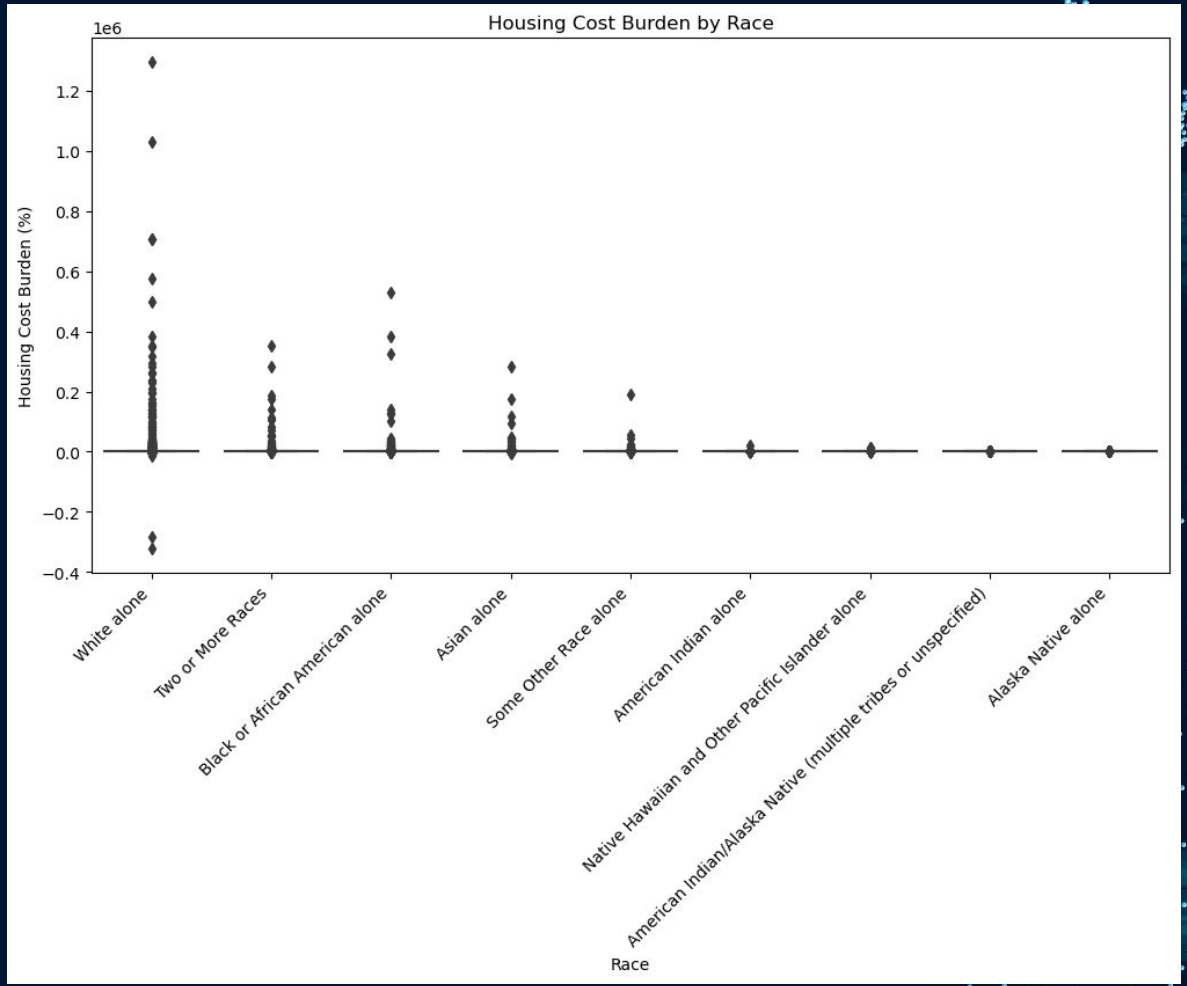
Code JupyterLab Python 3 (ipykernel)

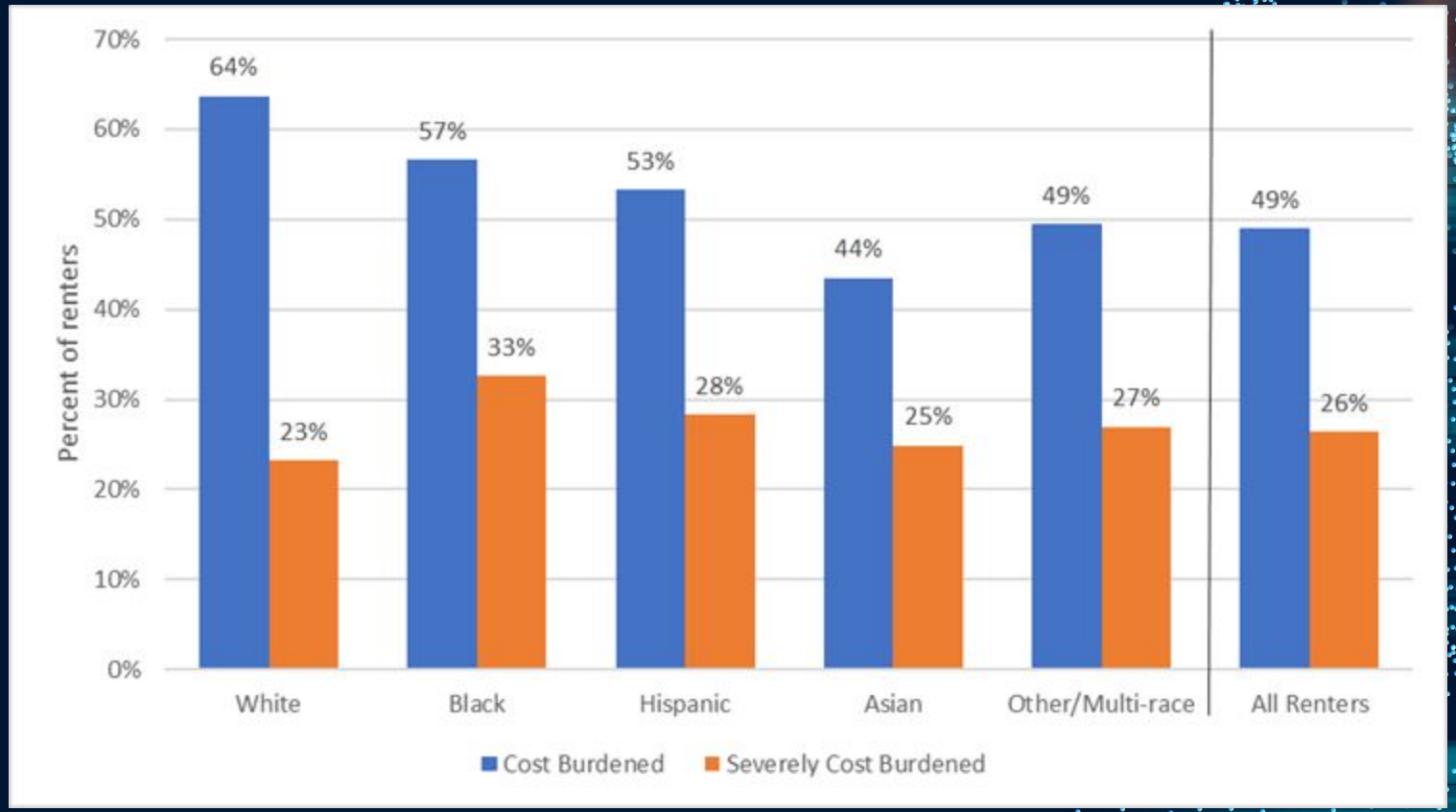
```
# Load the cleaned dataset with your selected variables
df = pd.read_csv('HousingVsIncomeCleaned2.csv')

# Calculate Adjusted Income (inflation-adjusted household income)
df['ADJUSTED_INCOME'] = df['HINCP'] * (df['ADJINC'] / 1000000)

# Total Monthly Housing Cost (Rent + Mortgage Payment)
df['HOUSING_COST'] = df['RNTP'] + df['MRGP']

# Calculate Housing Cost Burden (Annual Housing Cost as % of Income)
df['COST_BURDEN'] = (df['HOUSING_COST'] * 12 / df['ADJUSTED_INCOME']) * 100

#  Classify Cost Burden into Categories
df['BURDEN_CATEGORY'] = pd.cut(
    df['COST_BURDEN'],
    bins=[0, 30, 50, float('inf')],
    labels=['Affordable (<30%)', 'Burdened (30-49%)', 'Severely Burdened (50%+)']
)

# Add Race Mapping
race_lookup = {
    1: 'White alone',
    2: 'Black or African American alone',
    3: 'American Indian alone',
    4: 'Alaska Native alone',
    5: 'American Indian/Alaska Native (multiple tribes or unspecified)',
    6: 'Asian alone',
    7: 'Native Hawaiian and Other Pacific Islander alone',
    8: 'Some Other Race alone',
    9: 'Two or More Races'
}
df['RACE'] = df['RAC1P'].map(race_lookup)

# Save updated dataset (with burden & race names)
df.to_csv('acs_with_cost_burden_and_race.csv', index=False)

# Visualization Section

#  Bar Chart - Number of Households by Burden Level
plt.figure(figsize=(8,5))
sns.countplot(x='BURDEN_CATEGORY', data=df, palette='Set2')
plt.title('Number of Households by Housing Cost Burden')
plt.ylabel('Number of Households')
plt.xlabel('Cost Burden Category')
plt.show()

#  Pie Chart - Proportion of Households by Burden Level
df['BURDEN_CATEGORY'].value_counts().plot(
    kind='pie',
    autopct='%1.1f%%',
    startangle=140,
    colors=['#66c2a5', '#fc8d62', '#8da0cb']
)
plt.title('Proportion of Households by Housing Cost Burden')
plt.ylabel('')
plt.show()

#  Histogram - Distribution of Cost Burden
plt.figure(figsize=(10,6))
sns.histplot(df['COST_BURDEN'], bins=30, kde=True)
plt.axvline(30, color='red', linestyle='--', label='30% Burdened Threshold')
plt.axvline(50, color='orange', linestyle='--', label='50% Severely Burdened Threshold')
plt.legend()
plt.title('Distribution of Housing Cost Burden (%)')
```

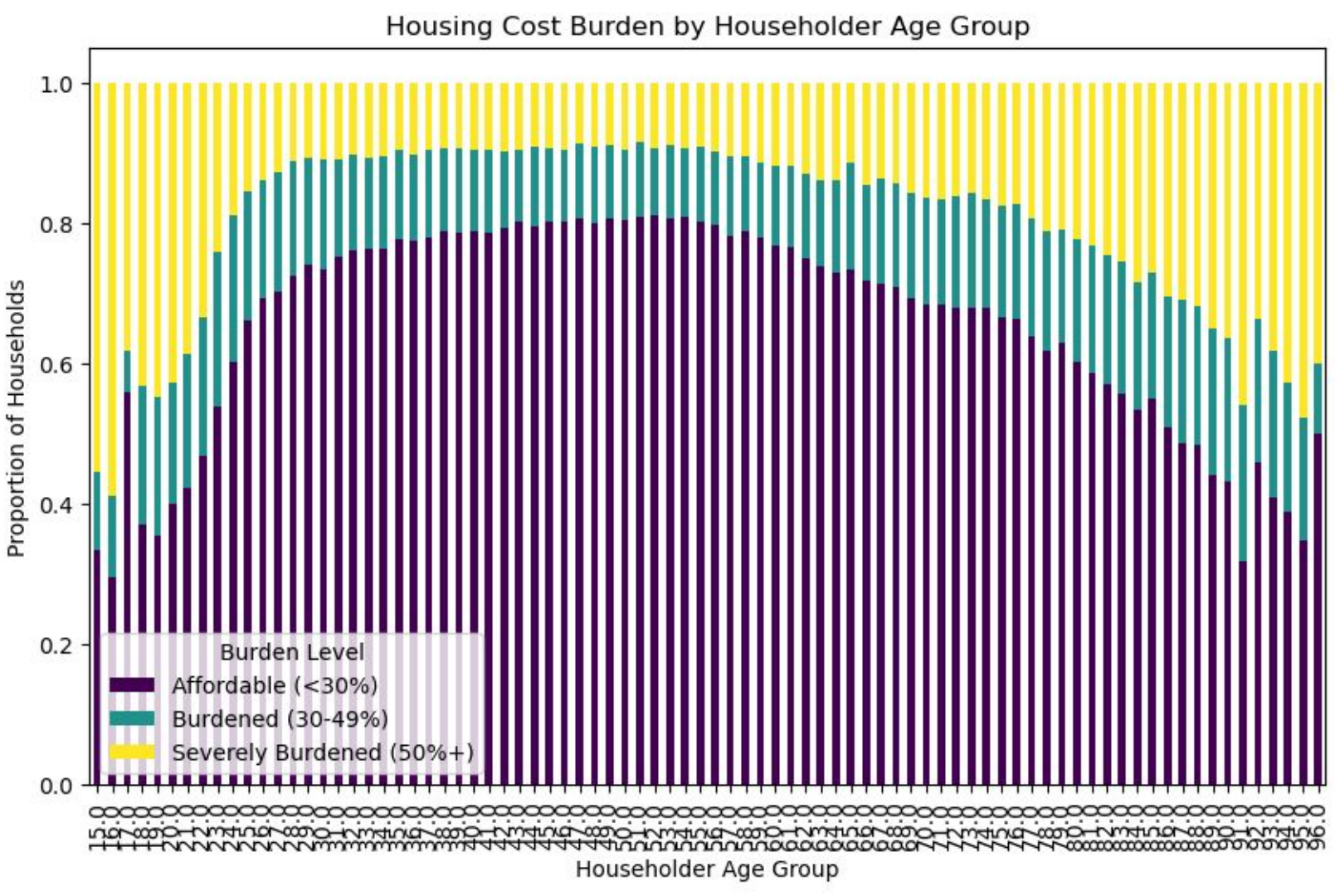
```
    9: 'Two or More Races'
}
df['RACE'] = df['RAC1P'].map(race_lookup)

# Save updated dataset (with burden & race names)
df.to_csv('acs_with_cost_burden_and_race.csv', index=False)

# Visualization Section

#  Bar Chart — Number of Households by Burden Level
plt.figure(figsize=(8,5))
sns.countplot(x='BURDEN_CATEGORY', data=df, palette='Set2')
plt.title('Number of Households by Housing Cost Burden')
plt.ylabel('Number of Households')
plt.xlabel('Cost Burden Category')
plt.show()

#  Pie Chart — Proportion of Households by Burden Level
df['BURDEN_CATEGORY'].value_counts().plot(
    kind='pie',
    autopct='%1.1f%%',
    startangle=140,
    colors=['#66c2a5', '#fc8d62', '#8da0cb']
)
plt.title('Proportion of Households by Housing Cost Burden')
plt.ylabel('')
plt.show()

#  Histogram — Distribution of Cost Burden
plt.figure(figsize=(10,6))
sns.histplot(df['COST_BURDEN'], bins=30, kde=True)
plt.axvline(30, color='red', linestyle='--', label='30% Burdened Threshold')
plt.axvline(50, color='orange', linestyle='--', label='50% Severely Burdened Threshold')
plt.legend()
plt.title('Distribution of Housing Cost Burden (%)')
plt.xlabel('Housing Cost Burden (%)')
plt.show()

#  Box Plot — Housing Cost Burden by Race (with Race Labels)
plt.figure(figsize=(12,6))
sns.boxplot(x='RACE', y='COST_BURDEN', data=df, palette='muted')
plt.xticks(rotation=45, ha='right')
plt.title('Housing Cost Burden by Race')
plt.xlabel('Race')
plt.ylabel('Housing Cost Burden (%)')
plt.show()

#  Stacked Bar Chart — Burden Level by Householder Age Group (AGEP)
burden_by_age = df.groupby('HHLDRAGEP')['BURDEN_CATEGORY'].value_counts(normalize=True).unstack(fill_v
burden_by_age.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10,6))
plt.title('Housing Cost Burden by Householder Age Group')
plt.ylabel('Proportion of Households')
plt.xlabel('Householder Age Group')
plt.legend(title='Burden Level')
plt.show()

# Save summary table
summary_table = df['BURDEN_CATEGORY'].value_counts(normalize=True).reset_index()
summary_table.columns = ['Burden Level', 'Proportion']
summary_table.to_csv('BurdenLevel.csv', index=False)

print(" Analysis complete! Results saved to 'CostBurdenAndRrace.csv' and 'BurdenLevel.csv'")
```


Number of Households by Housing Cost Burden
350000
300000
250000
200000
150000
100000
50000
0
Number of Households
Affordable (<30%)
Burdened (30-49%)
Severely Burdened (50%+)
Cost Burden Category


Proportion of Households by Housing Cost Burden
Severely Burdened (50%+)
Burdened (30-49%)
12.9%
13.3%
73.8%
Affordable (<30%)


Housing Cost Burden by Race
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
Housing Cost Burden (%)
White alone
Two or More Races
Black or African American alone
Asian alone
Some Other Race alone
American Indian alone
Native Hawaiian and Other Pacific Islander alone
American Indian/Alaska Native (multiple tribes or unspecified)
Alaska Native alone
Race


70%
60%
50%
40%
30%
20%
10%
0%
Percent of renters
64%
23%
57%
33%
53%
28%
44%
25%
49%
27%
49%
26%
White
Black
Hispanic
Asian
Other/Multi-race
All Renters
Cost Burdened
Severely Cost Burdened


Housing Cost Burden by Householder Age Group
Proportion of Households
1.0
0.8
0.6
0.4
0.2
0.0
Burden Level
Affordable (<30%)
Burdened (30-49%)
Severely Burdened (50%+)
Householder Age Group

What would I change?

Any errors or Issues?

Sources

- High Housing Costs are Consuming Household Incomes (Harvard Joint Center For Housing Studies
- Measuring Housing Affordability (PDF)
- High Rents are posing Financial Challenges (Urban Institute)
- National Low Income Housing Coalition
- Rising Cost Homeownership are Increasing Burdens
- Housing Costs Burden on Renters