Employment and the Labor Force

ANALYZING US CENSUS DATA IN PYTHON

Lee Hachadoorian
Asst. Professor of Instruction, Temple University

Employment Concepts

- **Labor Force**: People who are working or looking for work
- **Unemployed**: People unable to find work
- **Unemployment Rate**:

$$Unemployed/LaborForce$$

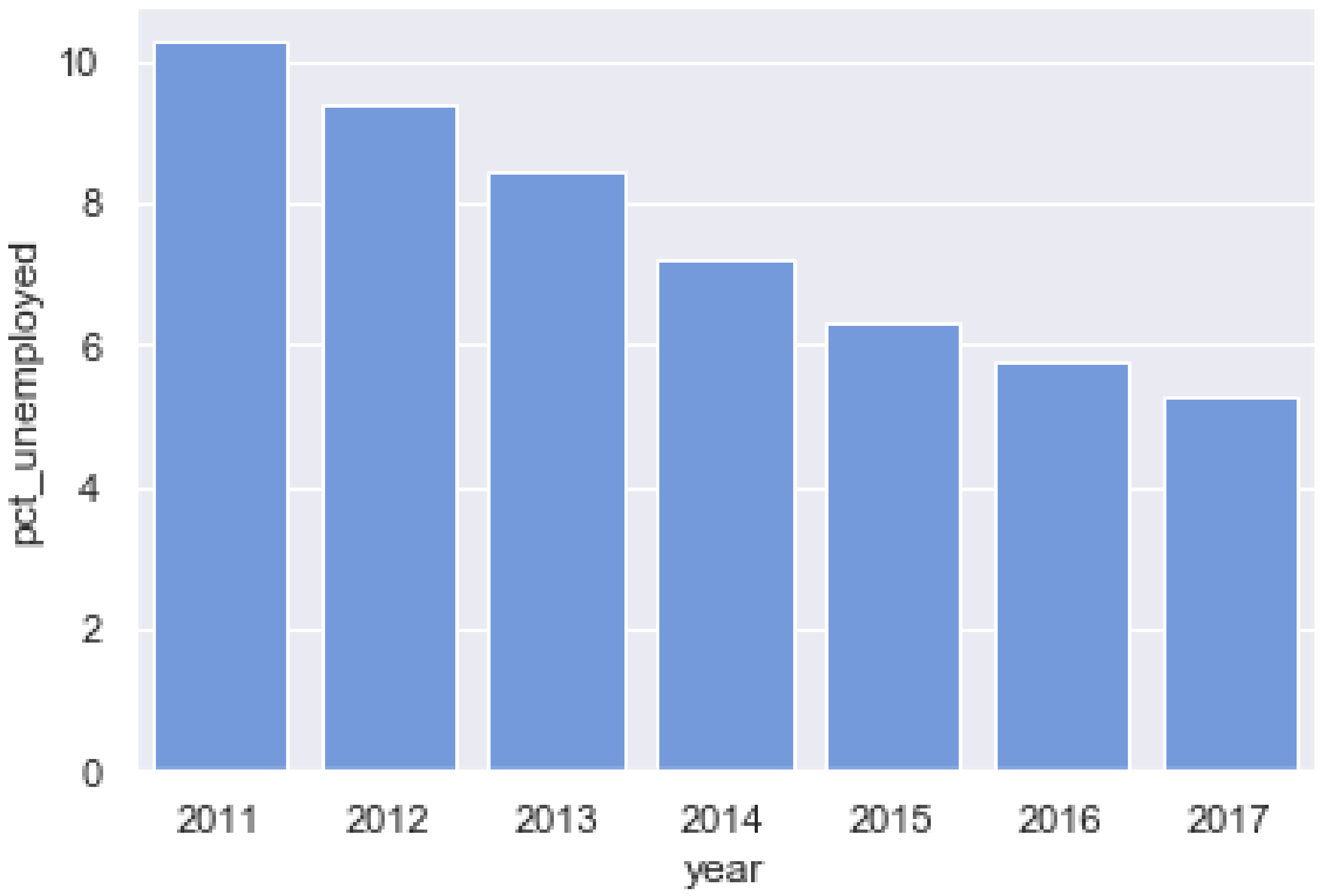
- **Labor Force Participation Rate**:

$$LaborForce/WorkingAgePop$$

Creating a Bar Plot

```
   year  pct_unemployed
0  2011       10.264992
0  2012        9.373092
0  2013        8.435212
0  2014        7.226895
0  2015        6.297886
0  2016        5.750313
0  2017        5.281027
```

```
sns.barplot(
  x = "year", y = "pct_unemployed",
  color = "cornflowerblue",
  data = employment)
```



pandas.melt

```
print(hispanic_unemployment)
```

```
   year  pct_hisp_male_25to54_unemp  pct_hisp_female_25to54_unemp
0  2011                    9.352638                     11.426135
0  2012                    8.062535                     10.751855
0  2013                    6.915451                      9.524808
0  2014                    5.724187                      8.285590
0  2015                    5.040303                      7.070101
0  2016                    4.568206                      6.521980
0  2017                    4.184646                      5.706956
```

pandas.melt

```
# Rename columns
col_rename = {"pct_hisp_male_25to54_unemp": "male",
              "pct_hisp_female_25to54_unemp": "female"}
hispanic_unemployment.rename(columns = col_rename, inplace = True)

# Melt data frame
tidy_unemp = hispanic_unemployment.melt(
        id_vars = "year",
        value_vars = ["male", "female"],
        var_name = "sex",
        value_name = "pct_unemployed")
```

pandas.melt

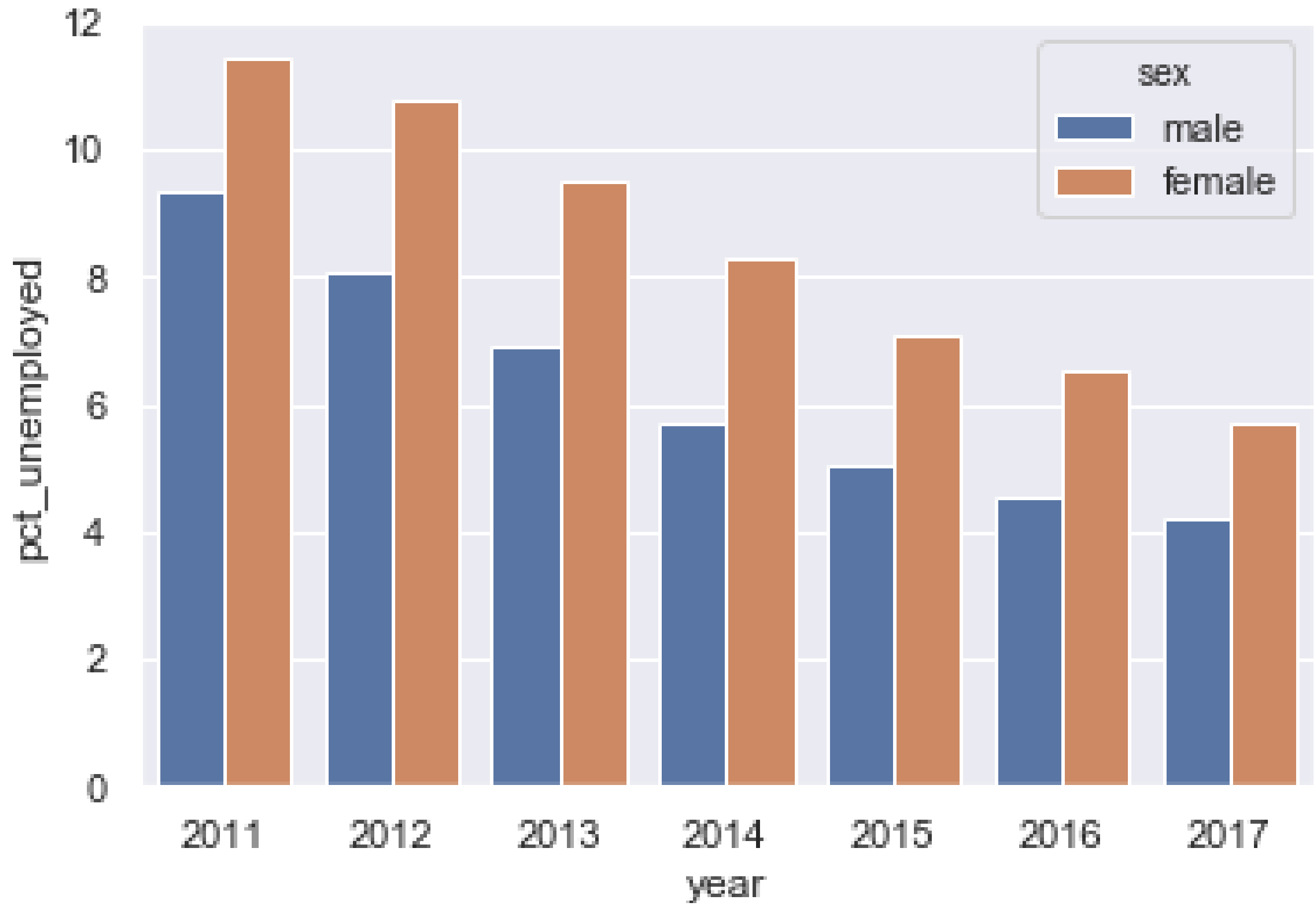
```
# Rename columns
col_rename = {"pct_hisp_male_25to54_unemp": "male",
              "pct_hisp_female_25to54_unemp": "female"}
hispanic_unemployment.rename(columns = col_rename, inplace = True)

# Melt data frame
tidy_unemp = hispanic_unemployment.melt(
        id_vars = "year",
        # value_vars = ["male", "female"],
        var_name = "sex",
        value_name = "pct_unemployed")
```

pandas.melt

```
    year     sex  pct_unemployed
0   2011    male        9.352638
1   2012    male        8.062535
2   2013    male        6.915451
3   2014    male        5.724187
4   2015    male        5.040303
5   2016    male        4.568206
6   2017    male        4.184646
7   2011  female       11.426135
8   2012  female       10.751855
9   2013  female        9.524808
10  2014  female        8.285590
11  2015  female        7.070101
12  2016  female        6.521980
13  2017  female        5.706956
```

Creating a Grouped Bar Chart

```
sns.barplot(x = "year", y = "pct_unemployed", hue = "sex",
            data = tidy_unemp)
```



Let's practice!

ANALYZING US CENSUS DATA IN PYTHON

Commuting Patterns

ANALYZING US CENSUS DATA IN PYTHON

Lee Hachadoorian

Asst. Professor of Instruction, Temple University

Commuting Tables

Commuting Subjects

- Means of transportation (car, public transit, etc.)
- Travel time
- Time leaving for/arriving at work

Commuting Geographies

- Residence: where people sleep
- Workplace: where people work; can use to determine workforce population for county, tract, etc.

Congestion Pricing in New York City

- Currently being debated in NYC (early 2019)
- Previous attempt failed (2007)
- Concerns over cost for low- and middle-income households

[1] Photo by Brian Jeffery Beggerly (CC BY 2.0)

Table B08519: Means Of Transportation To Work By Workers' Earnings In The Past 12 Months (In 2017 Inflation-Adjusted Dollars) For Workplace Geography

```
Total
    $1 to $9,999 or loss
    $10,000 to $14,999
    $15,000 to $24,999
    $25,000 to $34,999
    $35,000 to $49,999
    $50,000 to $64,999
    $65,000 to $74,999
    $75,000 or more
Car truck or van - drove alone
    <repeat income categories>
Car truck or van - carpooled
    <repeat income categories>
Public transportation (excluding taxicab)
    <repeat income categories>
etc...
```

API Response

```
print(r.json())
```

```
[['B08519_011E', 'B08519_012E', 'B08519_013E', 'B08519_014E', 'B08519_015E',
  'B08519_016E', 'B08519_017E', 'B08519_018E', 'B08519_020E', 'B08519_021E',
  ...
  'B08519_061E', 'B08519_062E', 'B08519_063E', 'state', 'county'],
 ['10927', '9172', '19659', '22110', '32287',
  '32977', '15693', '106972', '3663', '2518',
  ...
  '7457', '2664', '20684', '36', '061']]
```

Reshaping the Data

```python
# Read data row into list
data_row = r.json()[1][:-2]

# Break data row into list of lists
iter_len = 8
data = [data_row[i:i+iter_len] for i in range(0, len(data_row), iter_len)]

print(data)
```

```
[['10927', '9172', '19659', '22110', '32287', '32977', '15693', '106972'],
['3663', '2518', '5484', '5625', '8028', '7990', '3369', '22958'],
['139358', '97178', '200514', '184510', '255491', '240973', '116673', '700808'],
['16743', '9117', '15900', '13710', '17442', '20206', '10370', '85879'], ...]
```

Constructing the Data Frame

```
# Define row names and column names
modes = ["drove_alone", "carpooled", "public", "walked", "taxi",
         "worked_at_home"]

incomes = ["0k", "10k", "15k", "25k", "35k", "50k", "65k", "75k"]

# Create data frame
manhattan = pd.DataFrame(data=data, index=modes, columns=incomes)
manhattan = manhattan.astype(int)
```

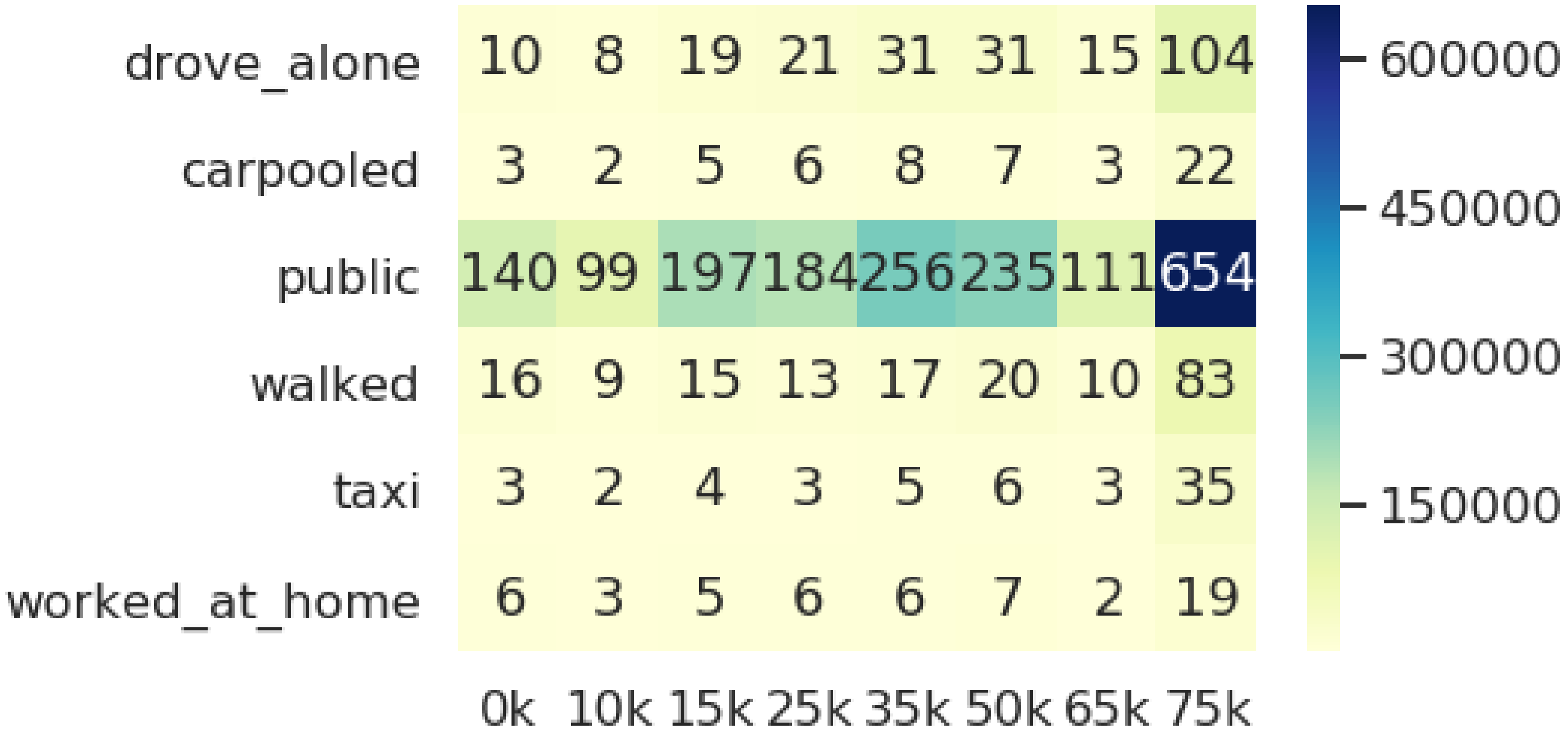
Constructing the Data Frame

```
print(manhattan)
```

```
                   0k     10k     15k  ...     50k     65k     75k
drove_alone     10716    8965   19294  ...   31502   15519  104078
carpooled        3740    2451    5852  ...    7994    3438   22625
public         140957   99474  197241  ...  235158  111959  654800
walked          16795    9045   15451  ...   20704   10663   83681
taxi             3201    2209    4515  ...    6551    3029   35572
worked_at_home   6854    3885    5489  ...    7776    2809   19598

[6 rows x 8 columns]
```

Constructing the Heatmap

```
# Create heatmap of commuters by mode by income
sns.heatmap(manhattan, annot=manhattan // 1000, fmt="d", cmap="YlGnBu")
```



Let's practice!

ANALYZING US CENSUS DATA IN PYTHON

Migration

ANALYZING US CENSUS DATA IN PYTHON

Lee Hachadoorian

Asst. Professor of Instruction, Temple University

ACS Mobility Tables - Common Columns

Table names "B07xxx", generally with columns like these:

- Total living in area (current residence)
  - Same house 1 year ago (i.e. did not move)
  - Moved within county
  - Moved from a different county, same state
  - Moved from a different state
  - Moved from abroad

ACS Mobility Tables - Additional Features

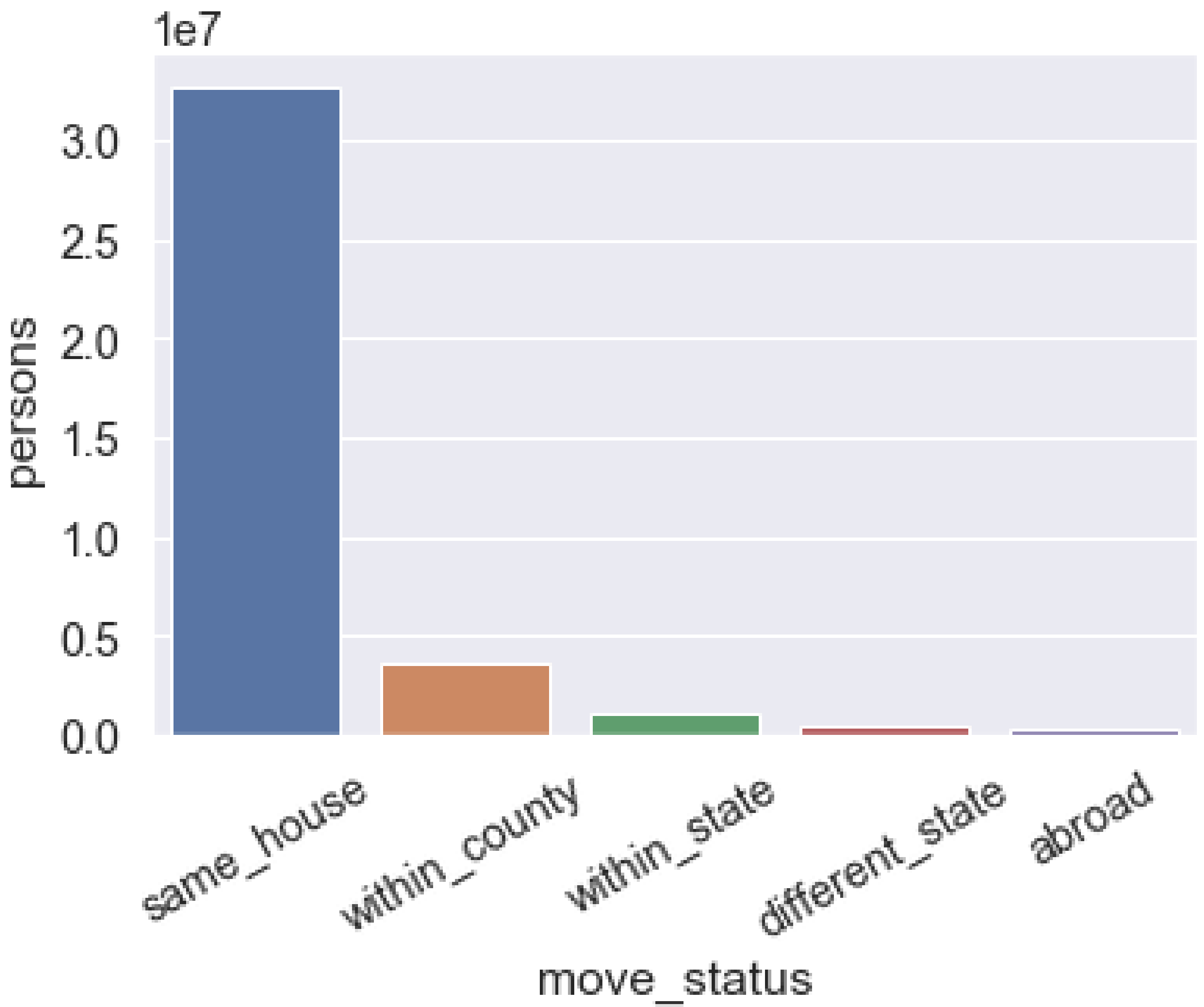
- Mobility crossed with:
  - Age
  - Educational Attainment
  - Income
  - Citizenship Status
  - etc.
- Tables based on residence 1 year ago
- Puerto Rico (e.g. B07001PR: Geographical Mobility in the Past Year by Age for Current Residence in Puerto Rico)

Going to California

```
print(to_cali_2016)
```

```
     move_status   persons
0     same_house  32740745
1  within_county   3581323
2   within_state   1062756
3 different_state   501384
4         abroad    305148
```

```
sns.barplot(x = "move_status",
            y = "persons",
            data = to_cali_2016)
```



Migration Flows

Table 1. State-to-State Migration Flows[1]: 2016										
Dataset: 2016 American Community Survey 1-Year Estimates										
Universe: Population 1 year and over										
	Population 1 year and over		Same house 1 year ago		Same state of residence 1 year ago		Different state of residence 1 year ago			
							Total		Alabama	
Current residence in --	Estimate	MOE	Estimate	MOE	Estimate	MOE	Estimate	MOE	Estimate	MOE
United States[2]	319,361,956	+/- 30,974	272,660,098	+/- 208,903	36,952,658	+/- 198,593	7,552,536	+/- 73,712	99,892	+/- 7,271
Alabama	4,810,126	+/- 3,913	4,141,850	+/- 18,249	529,994	+/- 17,409	122,220	+/- 9,811	N/A	N/A
Alaska	731,760	+/- 1,282	593,897	+/- 7,921	100,004	+/- 6,403	31,300	+/- 3,641	423	+/- 337
Arizona	6,851,836	+/- 4,173	5,586,753	+/- 30,225	938,077	+/- 26,500	273,257	+/- 14,805	894	+/- 543
Arkansas	2,949,650	+/- 3,135	2,484,705	+/- 17,957	384,811	+/- 15,832	71,083	+/- 6,657	2,057	+/- 1,295
California	38,783,436	+/- 10,247	33,594,813	+/- 62,303	4,337,251	+/- 59,356	514,758	+/- 19,678	3,045	+/- 1,187
Colorado	5,476,928	+/- 3,255	4,466,067	+/- 27,786	754,712	+/- 24,804	223,260	+/- 12,315	2,328	+/- 1,640
Connecticut	3,541,758	+/- 2,975	3,116,440	+/- 16,535	323,316	+/- 14,648	75,586	+/- 6,930	1,102	+/- 930
Delaware	942,073	+/- 1,325	817,779	+/- 9,121	85,147	+/- 8,425	33,400	+/- 3,275	148	+/- 253
District of Columbia	672,022	+/- 1,765	538,547	+/- 7,335	64,166	+/- 6,120	58,154	+/- 4,335	140	+/- 161
Florida	20,401,575	+/- 7,828	17,176,492	+/- 49,542	2,387,227	+/- 43,437	605,018	+/- 21,606	11,353	+/- 2,540

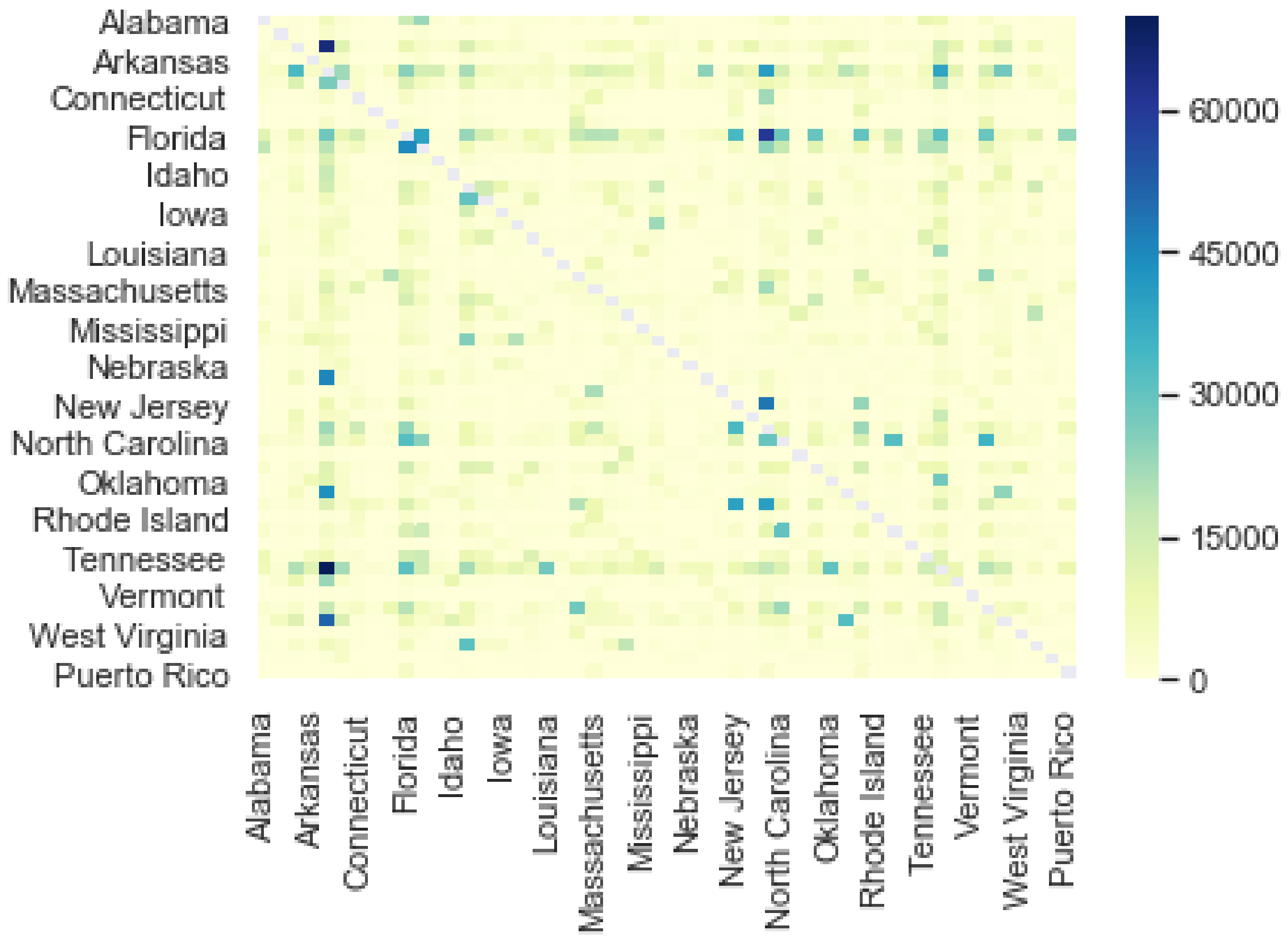
State-to-State Migration Matrix

```
print(state_to_state.head())
```

```
            Alabama  Alaska  Arizona  ...  Wisconsin  Wyoming  Puerto Rico
Alabama         NaN   576.0   1022.0  ...      874.0    539.0        335.0
Alaska        423.0     NaN   1176.0  ...      260.0    291.0        848.0
Arizona       894.0  1946.0      NaN  ...     6736.0    925.0       1462.0
Arkansas     2057.0   103.0    836.0  ...      539.0    178.0        857.0
California   3045.0  4206.0  33757.0  ...     7354.0   2674.0       1102.0
```

State-to-State Migration Heatmap

```
sns.heatmap(state_to_state, cmap="YlGnBu")
```



Let's practice!

ANALYZING US CENSUS DATA IN PYTHON

Is the Rent Too Damn High?

ANALYZING US CENSUS DATA IN PYTHON

Lee Hachadoorian

Asst. Professor of Instruction, Temple University

Definitions

Different ways of calculating rent:

- **Contract Rent**: Rent paid on a lease
- **Gross Rent**: Rent plus utilities; utilities may be included in contract rent on some leases, paid separately by the renter on other leases

Rent burden:

- **Rent Burden**: Paying 30% or more of household income in rent
- **Severe Rent Burden**: Paying 50% or more of household income in rent

Table B25074: HH Income By Gross Rent As a Percentage of HH Income in the Past 12 Months

```
Total
    Less than $10,000
        Less than 20.0 percent
        20.0 to 24.9 percent
        25.0 to 29.9 percent
        30.0 to 34.9 percent
        35.0 to 39.9 percent
        40.0 to 49.9 percent
        50.0 percent or more
        Not computed
    $10,000 to $19,999
    $20,000 to $34,999
    $35,000 to $49,999
    $50,000 to $74,999
    $75,000 to $99,999
    $100,000 or more
```

United States Rent Share of Income, ACS 2012-2016

```
total                                    42835169
inc_under_10k                             5558843
inc_under_10k_rent_under_20_pct             57052
inc_under_10k_rent_20_to_25_pct             58042
inc_under_10k_rent_25_to_30_pct            208806
inc_under_10k_rent_30_to_35_pct            177709
inc_under_10k_rent_35_to_40_pct            102565
inc_under_10k_rent_40_to_50_pct            150153
inc_under_10k_rent_over_50_pct            3381537
inc_under_10k_rent_not_computed           1422979
inc_10k_to_20k                            7027373
inc_10k_to_20k_rent_under_20_pct           213000
etc...
```

Calculating Rent Burden

```
print(rent.columns[10:19])
```

```
Index(['inc_10k_to_20k', 'inc_10k_to_20k_rent_under_20_pct',
    'inc_10k_to_20k_rent_20_to_25_pct', 'inc_10k_to_20k_rent_25_to_30_pct',
    'inc_10k_to_20k_rent_30_to_35_pct', 'inc_10k_to_20k_rent_35_to_40_pct',
    'inc_10k_to_20k_rent_40_to_50_pct', 'inc_10k_to_20k_rent_over_50_pct',
    'inc_10k_to_20k_rent_not_computed'],
  dtype='object')
```

Calculating Rent Burden

```
rent["inc_10k_to_20k_rent_burden"] = 100 * (
    rent["inc_10k_to_20k_rent_30_to_35_pct"] +
    rent["inc_10k_to_20k_rent_35_to_40_pct"] +
    rent["inc_10k_to_20k_rent_40_to_50_pct"] +
    rent["inc_10k_to_20k_rent_over_50_pct"]
    ) / (
    rent["inc_10k_to_20k"] -
    rent["inc_10k_to_20k_rent_not_computed"]
    )
```

Calculating Rent Burden

```
print(rent["inc_10k_to_20k_rent_burden"])
```

```
0    87.008024
Name: inc_10k_to_20k_rent_burden, dtype: float64
```

Calculating Rent Burden in a Loop

```
# Create list with income category part of column names
incomes = ["inc_under_10k", "inc_10k_to_20k", "inc_20k_to_35k",
           "inc_35k_to_50k", "inc_50k_to_75k", "inc_75k_to_100k",
           "inc_over_100k"]
```

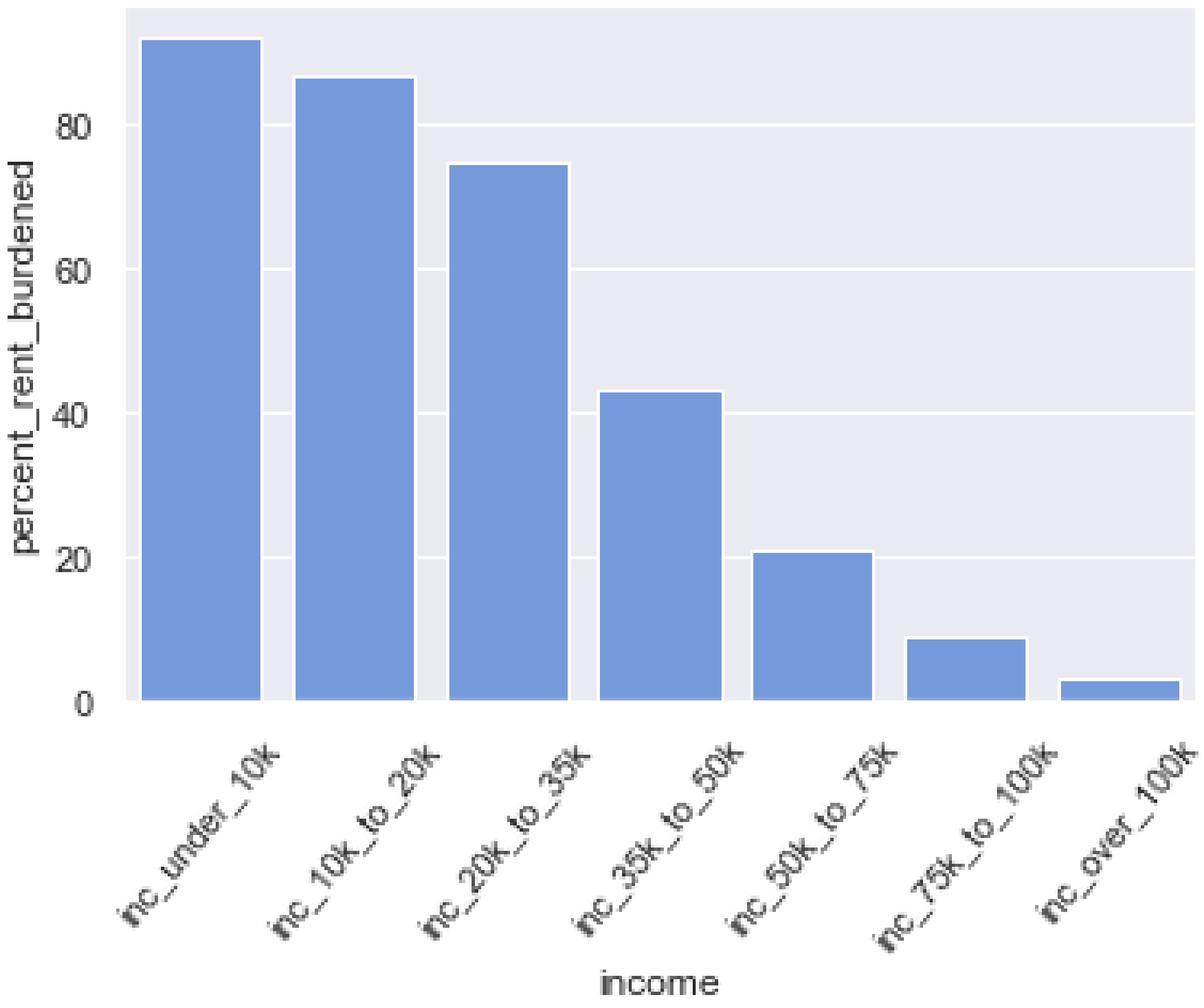
Calculating Rent Burden in a Loop

```
# Create new data frame with just the geography name
rent_burden = rent["name"]

# Loop over the list of income categories
for income in incomes:
    # Construct column names
    rent_burden[income] =
        100 * (rent[income + "_rent_30_to_35_pct"] +
        rent[income + "_rent_35_to_40_pct"] +
        rent[income + "_rent_40_to_50_pct"] +
        rent[income + "_rent_over_50_pct"]) / (
        rent[income] - rent[income + "_rent_not_computed"])
```

United States Rent Burden by Income Category

```
print(rent_burden.squeeze())
```

```
name                     United States
inc_under_10k                  92.1685
inc_10k_to_20k                  87.008
inc_20k_to_35k                 74.7448
inc_35k_to_50k                 43.0434
inc_50k_to_75k                 21.0937
inc_75k_to_100k                9.11853
inc_over_100k                  3.14882
Name: 0, dtype: object
```



Let's practice!

ANALYZING US CENSUS DATA IN PYTHON

Congratulations!

ANALYZING US CENSUS DATA IN PYTHON

Lee Hachadoorian

Asst. Professor of Instruction, Temple University

Decennial Census of Population and Housing

- Full count conducted every 10 years
- Covers core demographic topics
- Available for smallest geographies

American Community Survey

- Annual survey of 1.5% of households
- Covers a wide range of social and economic topics
- Available for 1-year and 5-year averages
- Pay attention to Margins of Error
- Limited availability for smallest geographies

Census Topics

Some topics we covered

- Race
- Hispanic Origin
- Employment and Labor Force
- Commuting
- Migration
- Home Value/Rent
- Health Insurance
- Computer/Internet Access

Some topics not covered

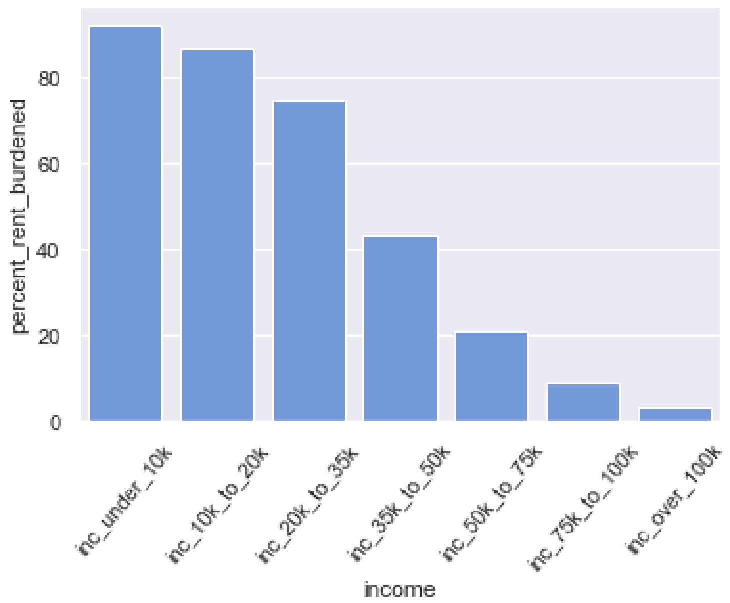
- Disability Status
- Veteran Status
- Industry and Occupation
- Poverty
- School Enrollment
- Grandparents as Caregivers
- Marital Status
- Language Spoken at Home

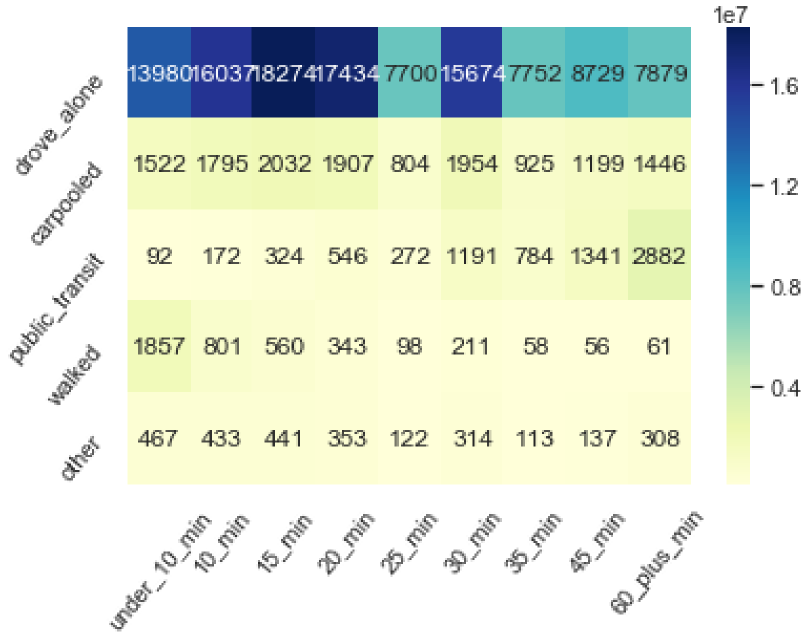
pandas

- Data aggregation with `groupby()`
- Joining data with `merge()`
- Tidy data: `pivot()` and `melt()`

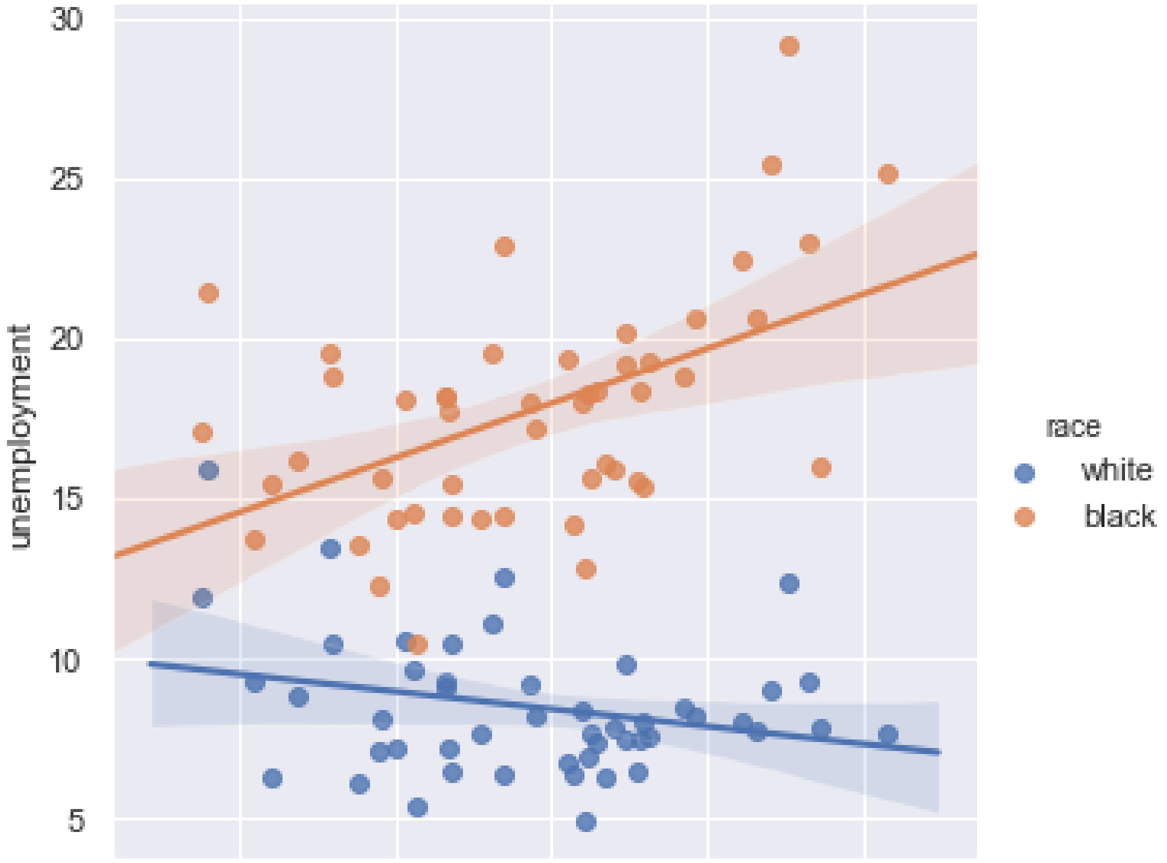
- pandas Foundations
- Manipulating DataFrames with pandas
- Merging DataFrames with pandas

seaborn



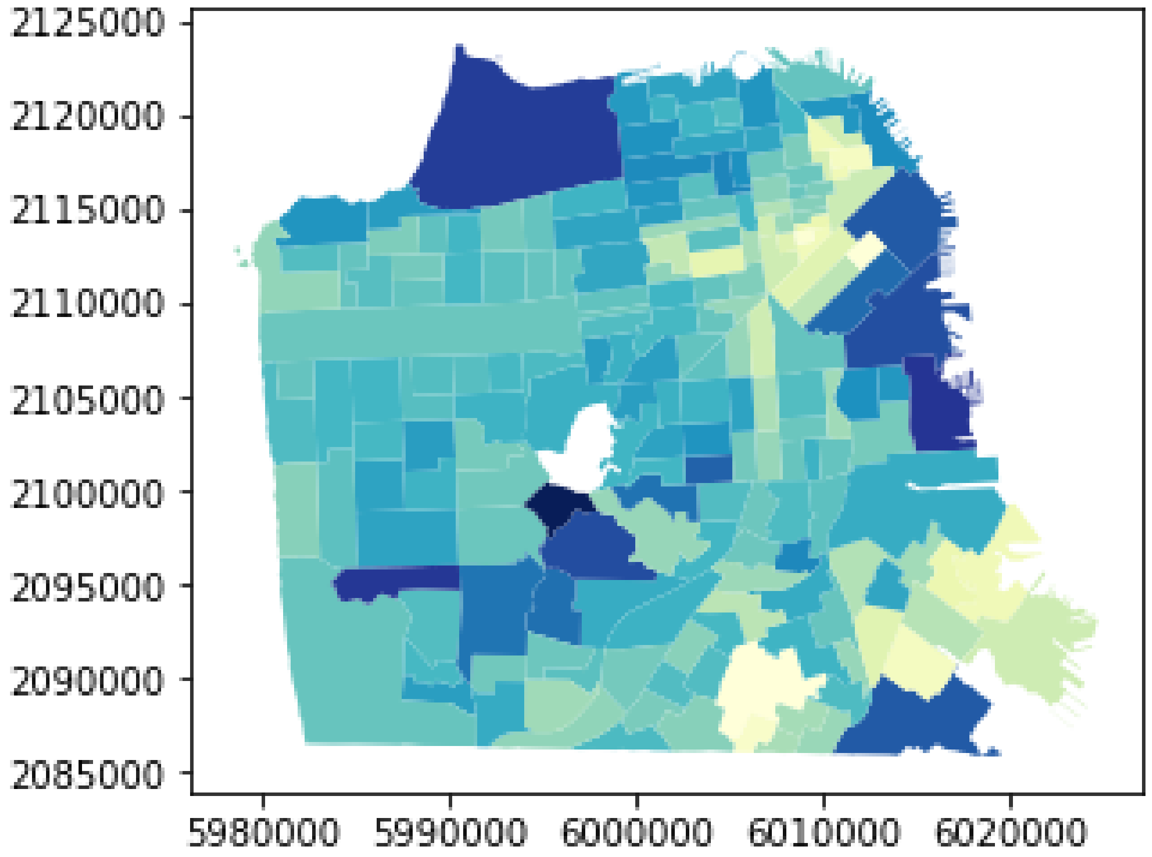


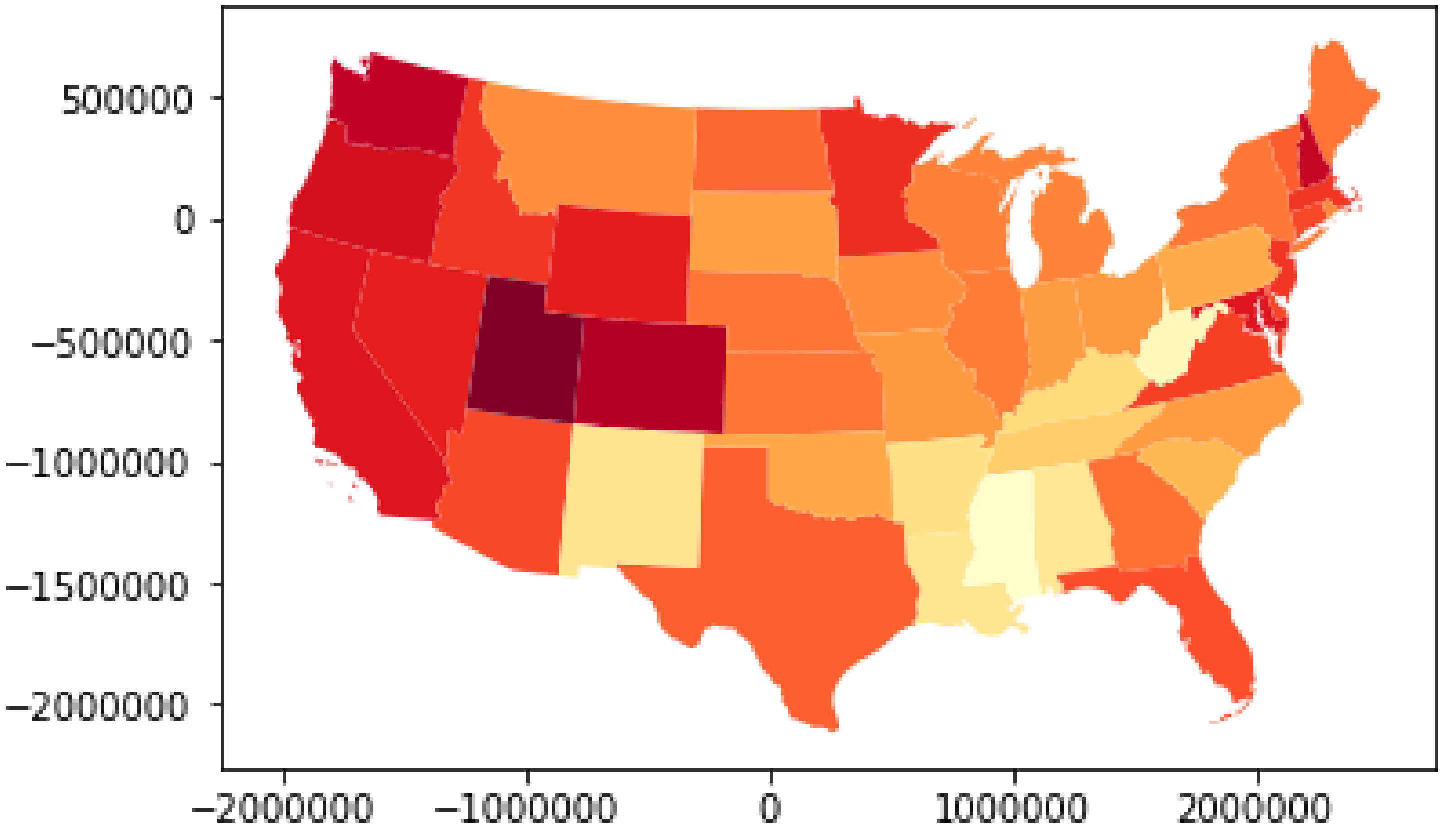
- Introduction to Data Visualization with Python
- Data Visualization with Seaborn



geopandas

- Working with Geospatial Data in Python
- Visualizing Geospatial Data in Python





Have fun exploring the Census!

ANALYZING US CENSUS DATA IN PYTHON