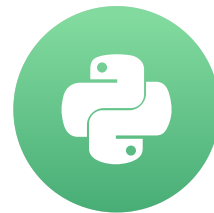


# Measuring Segregation: The Index of Dissimilarity

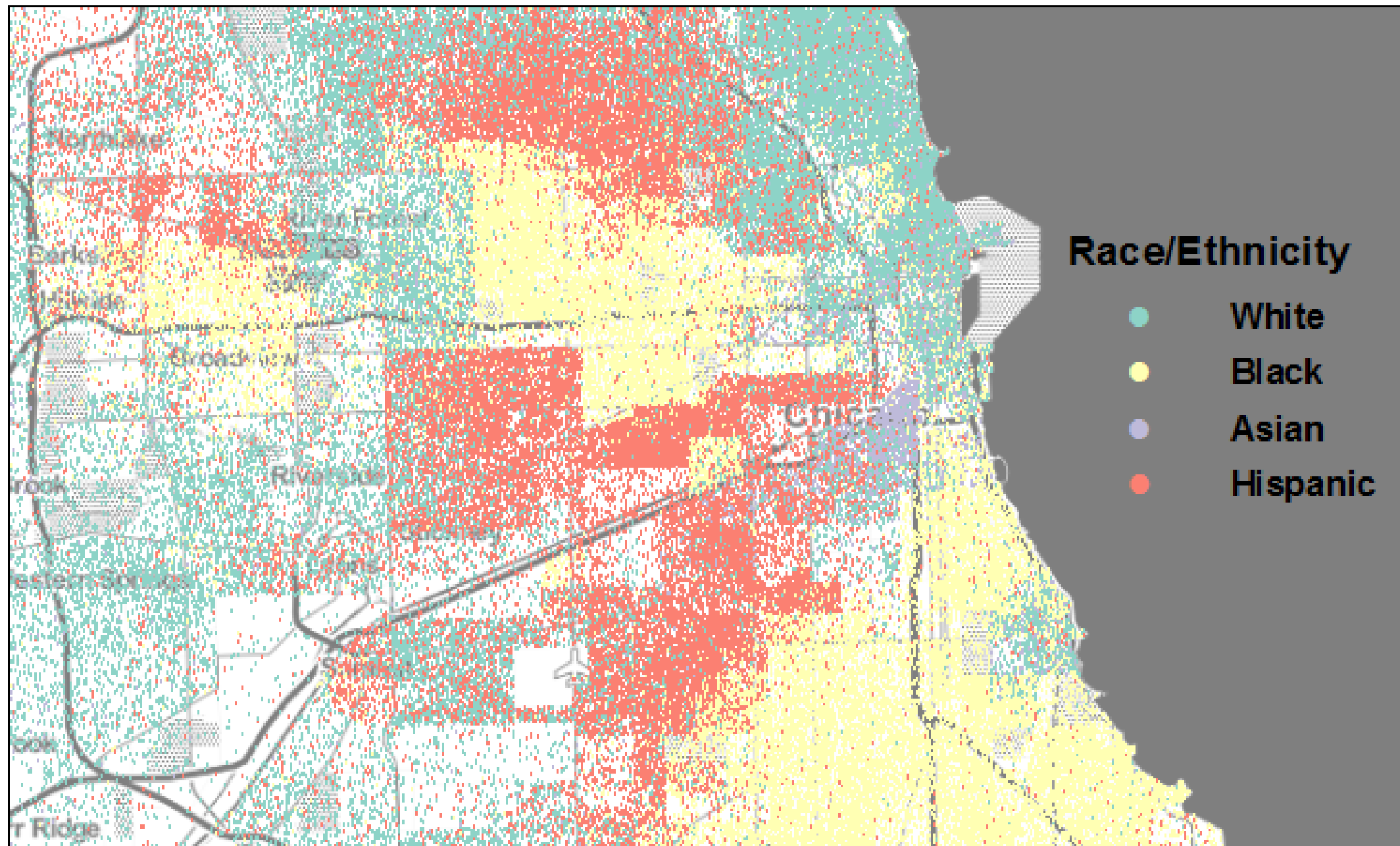
ANALYZING US CENSUS DATA IN PYTHON

**Lee Hachadoorian**

Asst. Professor of Instruction, Temple  
University



# What is Segregation?



# Index of Dissimilarity Formula

Given two groups A and B:

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \frac{1}{2} \sum_i \left| \frac{a_i}{A} - \frac{b_i}{B} \right|$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$
$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \frac{a_i}{A}$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \frac{b_i}{B}$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$



# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \left| \frac{a_i}{A} - \frac{b_i}{B} \right|$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \sum_i \left| \frac{a_i}{A} - \frac{b_i}{B} \right|$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \frac{1}{2} \sum_i \left| \frac{a_i}{A} - \frac{b_i}{B} \right|$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$

$$B = 400$$

# Index of Dissimilarity Formula

Given two groups A and B:

$$D = \frac{1}{2} \sum_i \left| \frac{a_i}{A} - \frac{b_i}{B} \right|$$

- $a_i$  = Small area Group A count
- $b_i$  = Small area Group B count
- $A$  = Large area Group A count
- $B$  = Large area Group B count

$a_1 = 10$ $b_1 = 200$	$a_2 = 20$ $b_2 = 100$
$a_3 = 30$ $b_3 = 50$	$a_4 = 40$ $b_4 = 50$

$$A = 100$$
$$B = 400$$

# Suitable Data

```
tracts.head()
```

```
   state county  tract  white  black
0     01     001  020100   1601    217
1     01     001  020200    844   1214
2     01     001  020300   2538    647
3     01     001  020400   4030    191
4     01     001  020500   8438   1418
```

Source: Table P5 - 2010 Decennial Census

- `white` = Nonhispanic White population
- `black` = Nonhispanic Black population

# Calculating the Index of Dissimilarity (D)

```
# Extract California tracts using state FIPS "06"
ca_tracts = tracts[tracts["state"] == "06"]

# Define convenience variables to hold column names
w = "white"
b = "black"
```

# Calculating the Index of Dissimilarity (D)

```
# Print the sum of Black population for all tracts in California  
print(ca_tracts[b].sum())
```

```
2163804
```

```
# Print the sum of White population for all tracts in California  
print(ca_tracts[w].sum())
```

```
14956253
```

# Calculating the Index of Dissimilarity (D)

$$D = \frac{1}{2} \sum_i \left| \frac{a_i}{A} - \frac{b_i}{B} \right|$$

```
# Calculate Index of Dissimilarity
print(0.5 * sum(abs(
    ca_tracts[w] / ca_tracts[w].sum() - ca_tracts[b] / ca_tracts[b].sum()
))))
```

```
0.6033425039167011
```



# Let's Practice!

ANALYZING US CENSUS DATA IN PYTHON

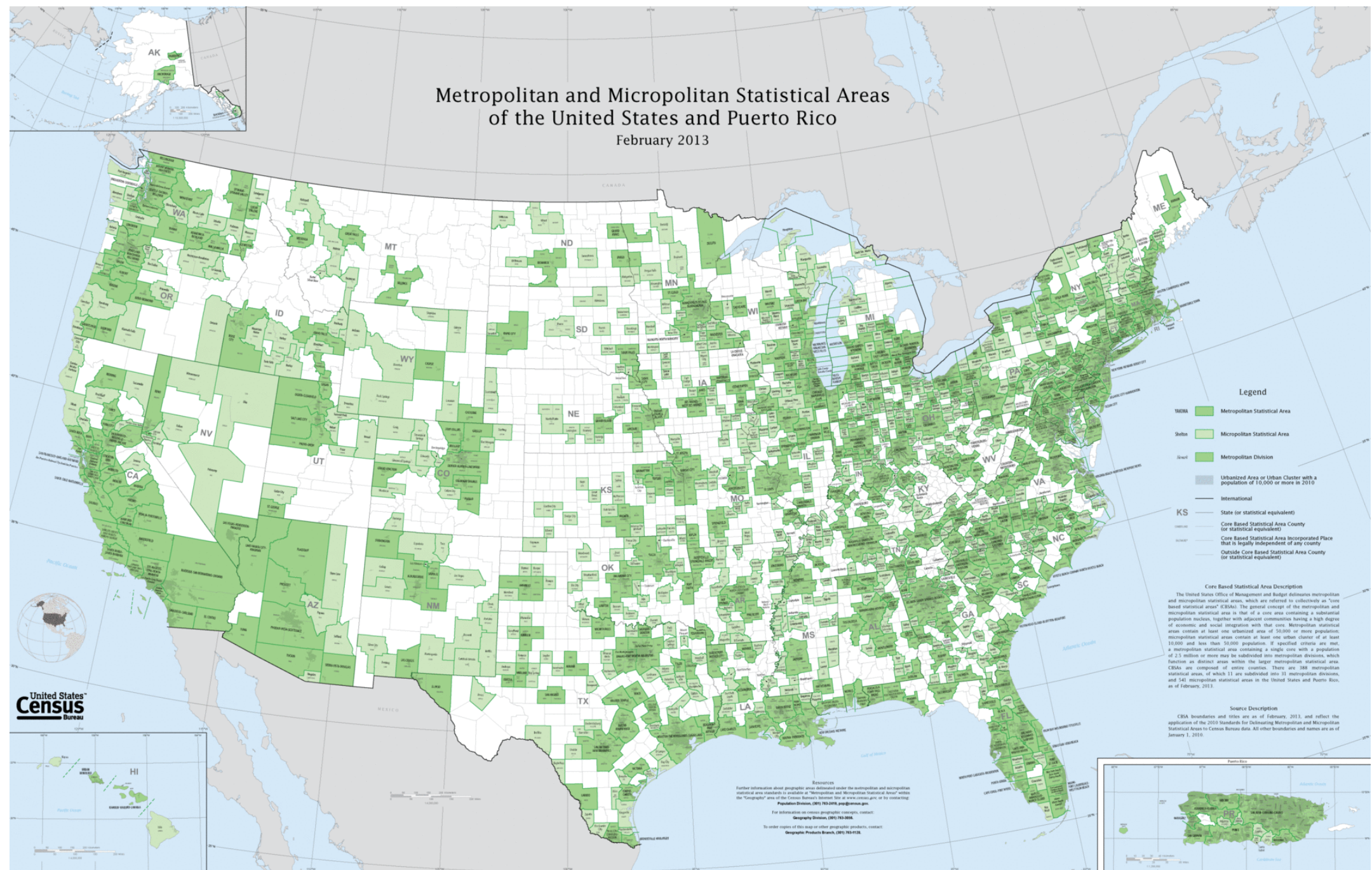
# Metropolitan Segregation

ANALYZING US CENSUS DATA IN PYTHON



**Lee Hachadoorian**

Asst. Professor of Instruction, Temple  
University



Source: United States Census Bureau

# Census API Request: Metro/Micropolitan Data

```
import requests

# Build base URL
HOST = "https://api.census.gov/data"
year = "2012"
dataset = "acs/acs5"
base_url = "/" .join([HOST, year, dataset])

# Specify requested variables
# B01001_001E = Total population (estimate)
# B03002_003E = Nonhispanic White population (estimate)
# B03002_004E = Nonhispanic Black population (estimate)
get_vars = ["NAME", "B01001_001E", "B03002_003E", "B03002_004E"]
```

# Census API Request: Metro/Micropolitan Data

```
# Specify requested variables
get_vars = ["NAME", "B01001_001E", "B03002_003E", "B03002_004E"]

# Create dictionary of predicates
predicates = {}
predicates["get"] = ",".join(get_vars)

# Requested geography
predicates["for"] = \
    "metropolitan statistical area/micropolitan statistical area:*
```

# Census API Request: Metro/Micropolitan Data

```
r = requests.get(base_url, params=predicates)
```

```
print(r.json()[ :5])
```

```
[['NAME', 'B01001_001E', 'B03002_003E', 'B03002_004E', 'metropolitan statistical area/mi  
['Adjuntas, PR Micro Area', '19458', '140', '0', '10260'],  
['Aguadilla-Isabela-San Sebastián, PR Metro Area', '305538', '5602', '231', '10380'],  
['Coamo, PR Micro Area', '71596', '228', '53', '17620'],  
['Fajardo, PR Metro Area', '70633', '543', '195', '21940']]
```



# Census API Request: Metro/Micropolitan Data

```
# Create user-friendly column names
col_names = ["name", "pop", "white", "black", "msa"]

# Load JSON response into data frame
msa = pd.DataFrame(columns=col_names, data=r.json()[1:])

# Cast count columns to int data type
msa[["pop", "white", "black"]] = msa[["pop", "white", "black"]].astype(int)
```

# Metropolitan Area Definition

	state	county	tract	white	black
0	01	001	020100	1601	217
1	01	001	020200	844	1214
2	01	001	020300	2538	647
3	01	001	020400	4030	191
4	01	001	020500	8438	1418

	msa	msa_name	county_name	state_name	state	county
0	10100	Aberdeen, SD	Brown County	South Dakota	46	013
1	10100	Aberdeen, SD	Edmunds County	South Dakota	46	045
2	10140	Aberdeen, WA	Grays Harbor County	Washington	53	027
3	10180	Abilene, TX	Callahan County	Texas	48	059
4	10180	Abilene, TX	Jones County	Texas	48	253



# Pandas Merge Method

```
import pandas as pd
```

```
# Join data frames on matching columns  
tracts_with_msa_id = pd.merge(...)
```

# Pandas Merge Method

```
import pandas as pd
```

```
# Join data frames on matching columns  
tracts_with_msa_id = pd.merge(tracts, msa_def, ...)
```

# Pandas Merge Method

```
import pandas as pd
```

```
# Join data frames on matching columns
tracts_with_msa_id = pd.merge(tracts, msa_def,
                              left_on = ["state", "county"], right_on = ["state", "county"])
```

```
# Alternative when column names are the same
tracts_with_msa_id = pd.merge(tracts, msa_def, on = ["state", "county"])
```

# Pandas Merge Method

```
# Data frame with state names  
st.head()
```

```
      state_name  
state  
01      Alabama  
02       Alaska  
04      Arizona  
05      Arkansas  
06    California
```

# Pandas Merge Method

```
# Join tracts and st data frames
tracts_st = pd.merge(tracts, st, left_on = "state", right_index = True)

tracts_st.head()
```

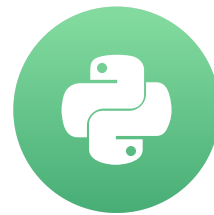
	state	county	tract	white	black	state_name
0	01	001	020100	1601	217	Alabama
1	01	001	020200	844	1214	Alabama
2	01	001	020300	2538	647	Alabama
3	01	001	020400	4030	191	Alabama

# Let's Practice

ANALYZING US CENSUS DATA IN PYTHON

# Segregation Impacts: Unemployment

ANALYZING US CENSUS DATA IN PYTHON



**Lee Hachadoorian**

Asst. Professor of Instruction, Temple  
University

# Deciphering ACS Subject Table IDs

[B|C]ssnnn[A-I]



# [B|C]ssnnn[A-I]

**B** or **C** = "Base Table" or "Collapsed Table"

B15002	C15002[A-I]
No schooling	Less than high school diploma
Nursery to 4th grade	High school grad, GED, or alt.
5th and 6th grade	Some college or associate's
7th and 8th grade	Bachelor's degree or higher
9th grade	
etc...	

# [B|C]ssnnn[A-I]

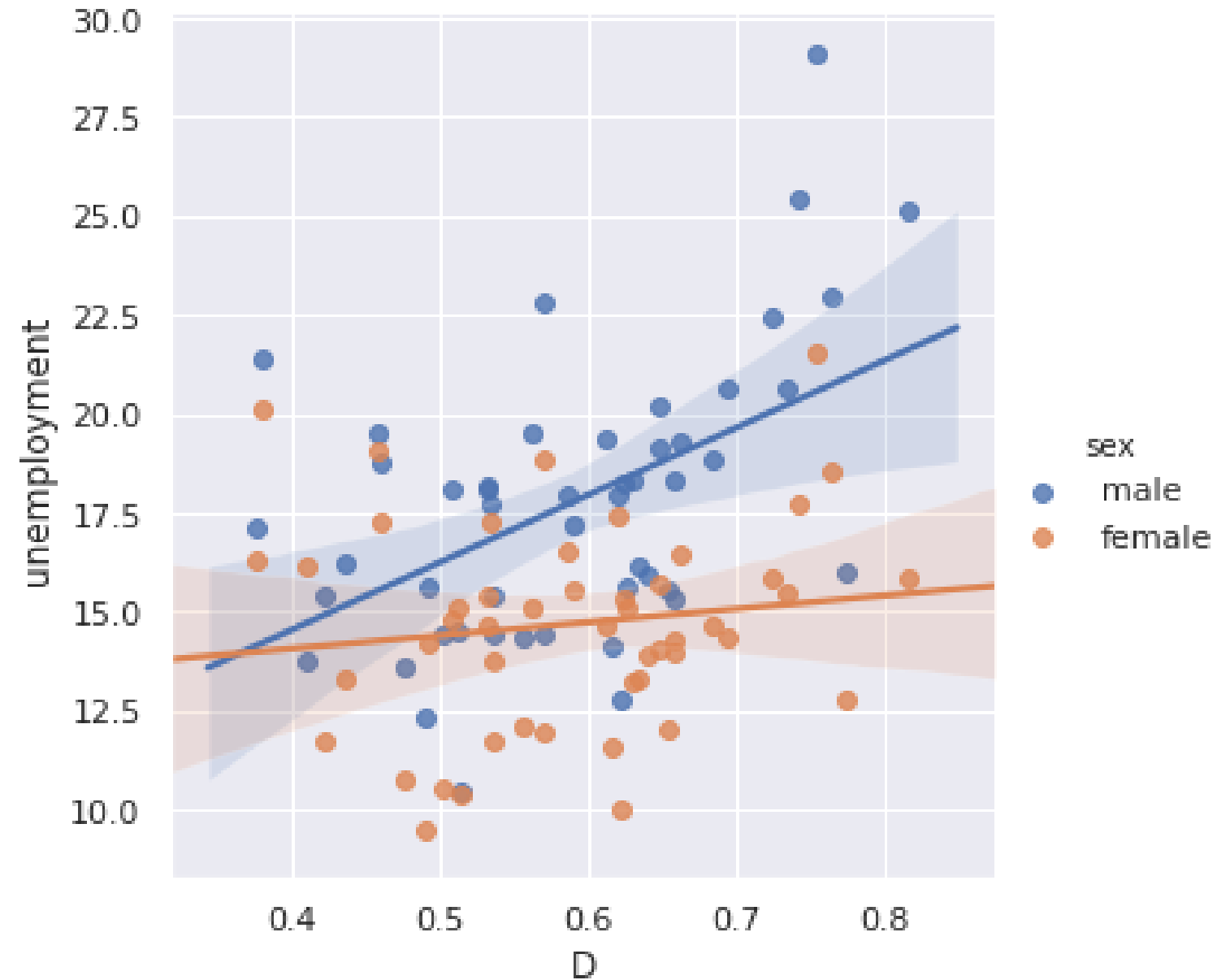
- A = White alone
- B = Black or African American Alone
- C = American Indian and Alaska Native Alone
- D = Asian Alone
- E = Native Hawaiian and Other Pacific Islander Alone
- F = Some Other Race Alone
- G = Two or More Races
- H = White Alone, Not Hispanic or Latino
- I = Hispanic or Latino

# [B|C]ssnnn[A-I]

- 01 = Age and Sex
- 02 = Race
- 03 = Hispanic or Latino Origin
- 05 = Foreign Born; Citizenship; Year of Entry; Nativity
- 15 = Educational Attainment
- 19 = Income (Households and Families)
- 23 = Employment Status; Work Experience; Labor Force

Source: <https://www.census.gov/programs-surveys/acs/guidance/which-data-tool/table-ids-explained.html>

# Comparing Segregation Impacts



# Tidy Data

Wide data frame: `msa_labor_force`

```
   msa  male_lf  female_lf
0  12060   400843   481425
1  25540   30656    35046
2  26420  231346   268923
3  26900   55943    71036
...
```

```
msa_labor_force.columns =
    ["msa", "male", "female"]
```

Tidy data frame: `tidy_msa_labor_force`

```
   msa  sex  labor_force
0  12060  male    400843
1  25540  male    30656
2  26420  male   231346
3  26900  male    55943
...
49 12060 female   481425
50 25540 female    35046
51 26420 female   268923
52 26900 female    71036
...
```

# pandas.melt

```
tidy_msa_labor_force = msa_labor_force.melt(  
    id_vars = ["msa"],  
    value_vars = ["male", "female"],  
    var_name = "sex",  
    value_name = "labor_force"  
)
```

# pandas.melt

```
tidy_msa_labor_force
```

```
   msa  sex  labor_force
0  12060 male    400843
1  25540 male     30656
2  26420 male    231346
3  26900 male     55943
...
49 12060 female   481425
50 25540 female    35046
51 26420 female   268923
52 26900 female    71036
...
```

# Let's Practice

ANALYZING US CENSUS DATA IN PYTHON

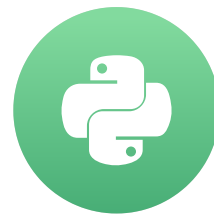


# Neighborhood Segregation Over Time

ANALYZING US CENSUS DATA IN PYTHON

**Lee Hachadoorian**

Asst. Professor of Instruction, Temple  
University

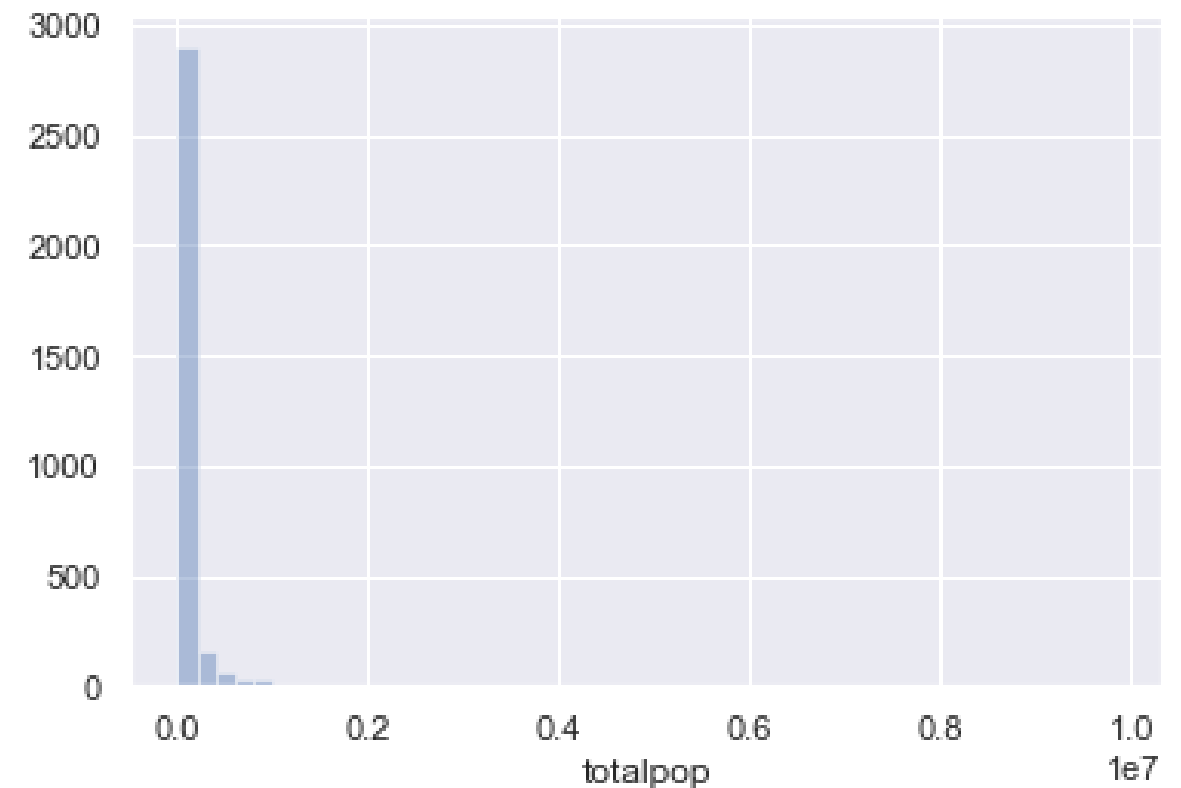


# Histograms

```
counties.head()
```

	totalpop	state	county
0	54571	01	001
1	22915	01	007
2	34215	01	017
3	25989	01	019
4	25833	01	025

```
sns.distplot(counties["totalpop"],  
             kde = False)
```

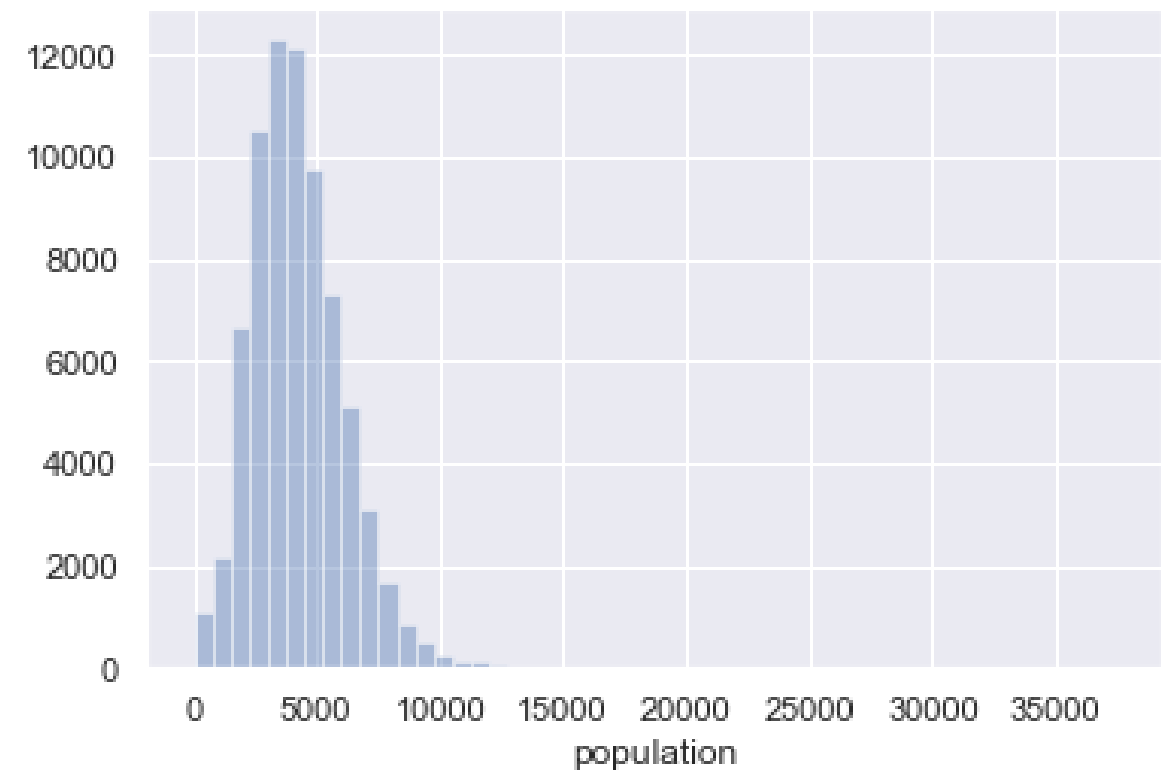


# Histograms

```
tracts.head()
```

	totalpop	state	county	tract
0	1912	01	001	020100
1	2170	01	001	020200
2	3373	01	001	020300
3	4386	01	001	020400
4	10766	01	001	020500

```
sns.distplot(tracts["totalpop"],  
             kde = False)
```





## RELEASE LOG

IPUMS provides census and survey data from around the world integrated across time and space. IPUMS integration and documentation makes it easy to study change, conduct comparative research, merge information across data types, and analyze individuals within family and community context. Data and services are available free of charge. [Learn more about IPUMS.](#)

# NHGIS vs. Census Bureau FTP

1. Historical data going back to 1790 (first United States Census)
2. GIS files for mapping Census data
3. Time series data for consistent geographic areas

# Let's Practice!

ANALYZING US CENSUS DATA IN PYTHON