```
# @title LATEST
import requests
import csv
import os
import time
import json

# Base URL of the Census API
base_url = "https://api.census.gov/data/"
api_key = 'c8d651e12d8f41c235a0e1edb4406f75c7d61072'

# Define the variables and corresponding tables as a dictionary
variables = {
    'S2503': 'acs/acs5/subject',
    'B17017': 'acs/acs5',
    'B25102': 'acs/acs5',
    'DP03': 'acs/acs5/profile',
    'S2506': 'acs/acs5/subject',
    'B19001': 'acs/acs5',
    'S1903': 'acs/acs5/subject'
}

# Define the years to download data for
years = range(2010, 2022)

# FIPS code for states
states = [
    '01', '02', '04', '05', '06', '08', '09', '10', '11', '12',
    '13', '15', '16', '17', '19', '20', '21', '22', '23', '24',
    '25', '26', '27', '28', '29', '30', '31', '32', '33', '34',
    '35', '36', '37', '38', '39', '40', '41', '42', '44', '45',
    '46', '47', '48', '49', '50', '51', '53', '54', '55', '56'
]
```

```
# Create a directory to store the downloaded data
output_dir = 'TRAIN_SET'
os.makedirs(output_dir, exist_ok=True)

# Iterate over the years
for year in years:
    # Iterate over the states
    for state in states:
        # Iterate over the variables
        for variable, table in variables.items():
            # Construct the request URL using the corresponding table
            request_url = f"{base_url}{year}/{table}?get=group({variable})&for=place:*&in=state:{state}&key={api_key}"

            # Make the request
            response = requests.get(request_url)

            # Check if request was successful
            if response.status_code == 200:
                try:
                    # Process the response
                    data = response.json()

                    # Check if response is not empty
                    if data:
                        # Save the data to a CSV file
                        filename = f'{variable}_survey_data_{year}_state_{state}.csv'
                        file_path = os.path.join(output_dir, filename)
                        with open(file_path, 'w', newline='', encoding='utf-8') as csvfile:
                            csv_writer = csv.writer(csvfile)
                            csv_writer.writerows(data)
                            print(f"Survey data for {variable} in {year} (State {state}) downloaded successfully as {filename}.")
                    else:
                        print(f"No data available for {variable} in {year} (State {state}).")
                except json.JSONDecodeError as e:
                    print(f"Error decoding JSON for {variable} in {year} (State {state}): {e}")
            else:
                print(f"Error downloading survey data for {variable} in {year} (State {state}): {response.status_code}")

            # Add a sleep to avoid hitting rate limits
            time.sleep(1.0)  # Adjust this value based on API rate limits



# @title Downlaoding the missing forms for 5 remainign states
#03,07,14,43,52
# @title LATEST
import requests
import csv
import os
import time
import json

# Base URL of the Census API
base_url = "https://api.census.gov/data/"
api_key = 'c8d651e12d8f41c235a0e1edb4406f75c7d61072'

# Define the variables and corresponding tables as a dictionary
variables = {
  'S2503': 'acs/acs5/subject',
  'B17017': 'acs/acs5',
  'B25102': 'acs/acs5',
  'DP03': 'acs/acs5/profile',
  'S2506': 'acs/acs5/subject',
    'B19001': 'acs/acs5',
    'S1903': 'acs/acs5/subject'
}

# Define the years to download data for
years = range(2010, 2022)

# FIPS code for states
states = ['03','07','14','43','52']



# Create a directory to store the downloaded data
output_dir = '/content/drive/MyDrive/HA_Train_Set'
#os.makedirs(output_dir, exist_ok=True)

# Iterate over the years
for year in years:
    # Iterate over the states
    for state in states:
        # Iterate over the variables
        for variable, table in variables.items():
            # Construct the request URL using the corresponding table
                                                                                request_url =
f"{base_url}{year}/{table}?get=group({variable})&for=place:*&in=state:{state}&key={api_ke
y}"

            # Make the request
            response = requests.get(request_url)

            # Check if request was successful
            if response.status_code == 200:
                try:
                    # Process the response
                    data = response.json()
```

```
          # Check if response is not empty
          if data:
            # Save the data to a CSV file
            filename = f'{variable}_survey_data_{year}_state_{state}.csv'
            file_path = os.path.join(output_dir, filename)
            with open(file_path, 'w', newline='', encoding='utf-8') as csvfile:
              csv_writer = csv.writer(csvfile)
              csv_writer.writerows(data)

            print(f"Survey data for {variable} in {year} (State {state}) downloaded successfully as {filename}.")
          else:
            print(f"No data available for {variable} in {year} (State {state}).")
        except json.JSONDecodeError as e:
          print(f"Error decoding JSON for {variable} in {year} (State {state}): {e}")
      else:
        print(f"Error downloading survey data for {variable} in {year} (State {state}): {response.status_code}")

      # Add a sleep to avoid hitting rate limits
      time.sleep(1.0)  # Adjust this value based on API rate limits
```

SAVING DATA INTO THE DRIVE

```
import shutil

# Specify the source directory where the files are saved in Colab
source_dir = '/content/TRAIN_SET'

# Specify the destination directory in your Google Drive
destination_dir = '/content/drive/My Drive/ml_Training_Set'

# Copy the files from the source directory to the destination directory
shutil.copytree(source_dir, destination_dir)
```

```
# Optionally, remove the source directory after copying
shutil.rmtree(source_dir)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Organize the data into the appropriate folders

```
# @title Organize the data into the appropriate folders
import os
import shutil

# Define the base directory to store the downloaded data
base_dir = '/content/drive/MyDrive/HA_Train_Set'

# Path to the folder containing the downloaded CSV files
train_set_dir = '/content/drive/MyDrive/ml_Training_Set'

# Iterate over the files in the train_Set directory
for root, _, files in os.walk(train_set_dir):
    for file in files:
        # Extract the year and variable from the file name
        #/content/drive/MyDrive/ml_Training_Set/B17017_survey_data_2010_state_01 (1).gsheet
        parts = file.split('_')
        year = parts[-3]
        variable = parts[0]

        # Create the directory structure if it doesn't exist
        year_dir = os.path.join(base_dir, year)
        variable_dir = os.path.join(year_dir, variable)
        os.makedirs(variable_dir, exist_ok=True)

        # Move the file to the appropriate directory
        src_path = os.path.join(root, file)
        dst_path = os.path.join(variable_dir, file)
        shutil.move(src_path, dst_path)

# @title Training Set info

data_dir = '/content/drive/MyDrive/HA_Train_Set'

# Iterate over the years
for year_folder in os.listdir(data_dir):
    year_path = os.path.join(data_dir, year_folder)
    if os.path.isdir(year_path):
        print(f"Year: {year_folder}")

        # Iterate over the variable categories
        for category_folder in os.listdir(year_path):
            category_path = os.path.join(year_path, category_folder)
            if os.path.isdir(category_path):
                print(f"\tSurvey: {category_folder}")

                # Iterate over the CSV files
                count=0
                for file_name in os.listdir(category_path):
                    if file_name.endswith('.csv'):
                        count=count+1
                        #print(count)
                print(f"\n")
                print(count)
```

Get the city list FROM S2503

```
# @title Get the city list FROM S2503
import os
import csv

cities = []
data = []
```

```
columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}

def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')

        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])



directory = '/content/drive/MyDrive/HA_Train_Set/2010/S2503'
for root, dirs, files in os.walk(directory):
    for file in files:
      file_path = os.path.join(root, file)
      if (file_path.lower().endswith('.csv')):
          print(file_path)
          get_cities(file_path)
```

Get the city list FROM B17017

```
# @title Get the city list FROM B17017
```

```
import os
import csv


columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}

def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')

        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])


directory = '/content/drive/MyDrive/HA_Train_Set/2010/B17017'
for root, dirs, files in os.walk(directory):
    for file in files:
      file_path = os.path.join(root, file)
      if (file_path.lower().endswith('.csv')):
          print(file_path)
          get_cities(file_path)
```

Get the city list FROM B19001

```
# @title Get the city list FROM B19001
import os
import csv


columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}

def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')

        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])


directory = '/content/drive/MyDrive/HA_Train_Set/2010/B19001'
for root, dirs, files in os.walk(directory):
    for file in files:
      file_path = os.path.join(root, file)
      if (file_path.lower().endswith('.csv')):
        print(file_path)
        get_cities(file_path)

```

Get the city list FROM B25102

```
# @title Get the city list FROM B25102
import os
import csv


columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}


def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')

        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])


directory = '/content/drive/MyDrive/HA_Train_Set/2010/B25102'
```

```
for root, dirs, files in os.walk(directory):
    for file in files:
        file_path = os.path.join(root, file)
        if (file_path.lower().endswith('.csv')):
            print(file_path)
            get_cities(file_path)
```

Get the city list FROM DPO3

```
# @title Get the city list FROM DPO3
import os
import csv


columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}

def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')

        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])
```

```
directory = '/content/drive/MyDrive/HA_Train_Set/2010/DP03'
for root, dirs, files in os.walk(directory):
    for file in files:
      file_path = os.path.join(root, file)
      if (file_path.lower().endswith('.csv')):
          print(file_path)
          get_cities(file_path)
```

Get the city list FROM S1903

```
# @title Get the city list FROM S1903
import os
import csv



columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}

def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')
        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])


directory = '/content/drive/MyDrive/HA_Train_Set/2010/S1903'
for root, dirs, files in os.walk(directory):
    for file in files:
      file_path = os.path.join(root, file)
      if (file_path.lower().endswith('.csv')):
          print(file_path)
          get_cities(file_path)


```

Get the city list FROM S2506

```
# @title Get the city list FROM S2506
import os
import csv


columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}


def get_cities(file_path):
    with open(file_path, 'r') as f:
        csvreader = csv.reader(f)
        header1 = next(csvreader)
        header2 = next(csvreader)  # Assuming the second row is also a header
        name_index = header1.index('NAME') if 'NAME' in header1 else header2.index('NAME')

        for row in csvreader:
            if row[name_index] not in cities:
                cities.append(row[name_index])



directory = '/content/drive/MyDrive/HA_Train_Set/2010/S2506'
for root, dirs, files in os.walk(directory):
    for file in files:
      file_path = os.path.join(root, file)
      if (file_path.lower().endswith('.csv')):
          print(file_path)
          get_cities(file_path)



print(cities)
import os
import csv

def extract_data_and_write(directory, output_file, columns_dictionary):
    data = {}  # Dictionary to store extracted data with city names as keys
    for root, dirs, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            if file.lower().endswith('.csv'):
                with open(file_path, 'r') as f:
                    csvreader = csv.DictReader(f)
                    #print(f"Extracting data from: {file_path}")
                    for row in csvreader:
                        name = row.get('NAME')  # Extract city name
                        if name:
                            #print(f"Extracted data for city: {name}")
                            if name not in data:
                                data[name] = {}  # Initialize data dictionary for city if not exists
                            for form, columns in columns_dictionary.items():
                                if form in row:
                                    if form not in data[name]:
                                        data[name][form] = {}  # Initialize form data dictionary if not exists
                                    for column in columns:
                                        if column in row:
                                              data[name][form][column] = row[column]  # Store column value for the city
                                            print(data[name][form][column])

    # Write extracted data to output CSV file
    with open(output_file, 'w', newline='') as csvfile:
        fieldnames = ['Name', 'Year'] + [col for cols in columns_dictionary.values() for col in cols]
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()

        for name, forms in data.items():
            for form, columns_data in forms.items():
                row_data = {'Name': name, 'Year': form}
                row_data.update(columns_data)  # Add form data to row
                writer.writerow(row_data)

# Example usage:
directory = '/content/drive/MyDrive/HA_Train_Set'
output_file = '/content/drive/MyDrive/HA_Train_Set/output.csv'   # Specify the desired output path here
columns_dictionary = {
    'S2503': ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E'],
    'B17017_2010_2014': ['B17017_002E'],
    'B25102': ['B25102_007E'],
    'DP03': ['DP03_0005E'],
    'S2506': ['S2506_C01_002E'],
    'B19001': ['B19001_011E'],
    'S1903': ['S1903_C02_016E']
}

extract_data_and_write(directory, output_file, columns_dictionary)

import os
import pandas as pd

def extract_data_and_write(directory, output_file, columns_dictionary):
    data = []  # List to store extracted data

    for root, dirs, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            if file_path.lower().endswith('.csv'):
                #print(f"Extracting data from: {file_path}")
                df = pd.read_csv(file_path, low_memory=False)  # Read CSV file into pandas
DataFrame with low_memory=False

                for form, columns in columns_dictionary.items():
                    print(f"Processing form: {form}, Columns: {columns}")
                    if form in df.columns:  # Check if form columns exist in DataFrame
                        selected_columns = ['NAME'] + columns  # Select columns to extract
                        extracted_data = df[selected_columns].copy()  # Extract selected columns
                        extracted_data['Year'] = os.path.basename(root)  # Add Year column
                        extracted_data.columns = ['Name', 'Year'] + columns  # Rename columns
                        data.append(extracted_data)  # Append extracted data to list
                        print(f"Extracted data length: {len(data)}")  # Print length of data list

    # Concatenate all extracted data into a single DataFrame
    extracted_data_df = pd.concat(data, ignore_index=True)
    # Write extracted data to output CSV file
    extracted_data_df.to_csv(output_file, index=False)
    print(f"Extracted data written to: {output_file}")

# Example usage:
directory = '/content/drive/MyDrive/HA_Train_Set'
output_file = '/content/drive/MyDrive/HA_Train_Set/output.csv'  # Specify the desired output
path here
columns_dictionary = {
}

extract_data_and_write(directory, output_file, columns_dictionary)

# @title LONG OUTPUT

import os
import pandas as pd

def extract_data_and_write(directory, output_file, columns_to_extract):
    data = []  # List to store extracted data

    for root, dirs, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            if file_path.lower().endswith('.csv'):
                print(f"Extracting data from: {file_path}")
                        df = pd.read_csv(file_path, low_memory=False)  # Read CSV file into pandas
DataFrame with low_memory=False

                for column in columns_to_extract:
                    if column in df.columns:  # Check if the column exists in DataFrame
                            extracted_data = df[['NAME', column]].copy()  # Extract selected column and
'NAME' column
                        parts = file_path.split('_')
                        year = parts[-3]
                        extracted_data['Year'] = year  # Add Year column
                        extracted_data.columns = ['Name', 'Year', column]  # Rename columns
                        data.append(extracted_data)  # Append extracted data to list
                        print(len(extracted_data))

    if data:  # Check if data is not empty
        # Concatenate all extracted data into a single DataFrame
        extracted_data_df = pd.concat(data, ignore_index=True)

        # Write extracted data to output CSV file
        extracted_data_df.to_csv(output_file, index=False)
        print(f"Extracted data written to: {output_file}")
    else:
        print("No data extracted. Check column names and file paths.")

# Example usage:
directory = '/content/drive/MyDrive/HA_Train_Set/'
output_file = '/content/drive/MyDrive/HA_Train_Set/output_final.csv'  # Specify the desired output path here
columns_to_extract = ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E', 'B17017_002E', 'B25102_007E', 'DP03_0005E', 'S2506_C01_002E', 'B19001_011E', 'S1903_C02_016E']

extract_data_and_write(directory, output_file, columns_to_extract)



import os
import pandas as pd

def extract_data_and_write(directory, output_file, columns_to_extract):
    data = []  # List to store extracted data

    for root, dirs, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            if file_path.lower().endswith('.csv'):
                print(f"Extracting data from: {file_path}")
                        df = pd.read_csv(file_path, low_memory=False)  # Read CSV file into pandas DataFrame with low_memory=False

                for column in columns_to_extract:
                    if column in df.columns:  # Check if the column exists in DataFrame
                        extracted_data = df[['NAME', 'Year', column]].copy()  # Extract selected columns and 'NAME' and 'Year' columns
                        parts = file_path.split('_')
                        year = parts[-3]
                        extracted_data['Year'] = year  # Update Year column
                        extracted_data.columns = ['Name', 'Year', column]  # Rename columns
                        data.append(extracted_data)  # Append extracted data to list

    if data:  # Check if data is not empty
        # Concatenate all extracted data into a single DataFrame
        extracted_data_df = pd.concat(data, ignore_index=True)

        # Reorder columns
        columns_order = ['Name', 'Year'] + columns_to_extract
        extracted_data_df = extracted_data_df[columns_order]

        # Write extracted data to output CSV file
        extracted_data_df.to_csv(output_file, index=False)
        print(f"Extracted data written to: {output_file}")
    else:
        print("No data extracted. Check column names and file paths.")

# Example usage:
directory = '/content/drive/MyDrive/HA_Train_Set/2010'
```

```
output_file = '/content/drive/MyDrive/HA_Train_Set/output5.csv'  # Specify the desired output path here
columns_to_extract = ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E', 'B17017_002E', 'B25102_007E', 'DP03_0005E', 'S2506_C01_002E', 'B19001_011E', 'S1903_C02_016E']

extract_data_and_write(directory, output_file, columns_to_extract)

def extract_data_and_write(directory, output_file, columns_to_extract):
    data = []  # List to store extracted data

    for root, dirs, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            if file_path.lower().endswith('.csv'):
                print(f"Extracting data from: {file_path}")
                         df = pd.read_csv(file_path, low_memory=False)  # Read CSV file into pandas DataFrame with low_memory=False

                for column in columns_to_extract:
                    if column in df.columns:  # Check if the column exists in DataFrame
                              extracted_data = df[['NAME', column]].copy()  # Extract selected column and 'NAME' column
                        parts = file_path.split('_')
                        year = parts[-3]
                        extracted_data['Year'] = year  # Add Year column
                        extracted_data.columns = ['Name', column, 'Year']  # Rename columns
                        data.append(extracted_data)  # Append extracted data to list
                        print(extracted_data)

    if data:  # Check if data is not empty
        # Concatenate all extracted data into a single DataFrame
        extracted_data_df = pd.concat(data, ignore_index=True)
        # Pivot the DataFrame to ensure 'Name' and 'Year' are the joint primary key
         extracted_data_df = extracted_data_df.pivot_table(index=['Name', 'Year'], columns=None, values=columns_to_extract, aggfunc='first').reset_index()

        # Write extracted data to output CSV file
        extracted_data_df.to_csv(output_file, index=False)
        print(f"Extracted data written to: {output_file}")
    else:
        print("No data extracted. Check column names and file paths.")

# Example usage:
directory = '/content/drive/MyDrive/HA_Train_Set/2010'
output_file = '/content/drive/MyDrive/HA_Train_Set/output_final2.csv'  # Specify the desired output path here
columns_to_extract = ['S2503_C01_001E', 'S2503_C01_002E', 'S2503_C01_014E', 'B17017_002E', 'B25102_007E', 'DP03_0005E', 'S2506_C01_002E', 'B19001_011E', 'S1903_C02_016E']

extract_data_and_write(directory, output_file, columns_to_extract)
```