Final Project: Clean Data

This script is meant to follow the Get_Data scipt. This tool removes fields from the demographic data and combines all demographic into one table. The demographic data is then joined to the spatial geometry on a census tracts level using ArcPy.

In [ ]:
```
import pandas as pd
import os
```

Census Data

Age

In [ ]:
```
age_csv = r"C:\Users\msong\Desktop\arc2proj\data\agesex\ACSST5Y2019.S0101_data
_with_overlays_2021-04-23T093838.csv"
cols =["GEO_ID",
       "NAME",
       "S0101_C01_001E", # estimate total pop
       "S0101_C01_026E", # pop 18+
       "S0101_C01_027E", # pop 21+
       "S0101_C01_028E", # pop 60+
       "S0101_C01_032E"  # median age
      ]
age_df = pd.read_csv (age_csv,
                      header=0,
                      usecols= cols,
                     )
```

In [ ]:
```
age_df = age_df.iloc[1: , :] # remove first row with heading descriptions
```

In [ ]:
```
# rename columns
new_cols = ["geo_id",
            "tract",
            "total_pop",
            "pop_over_18",
            "pop_over_21",
            "pop_over_60",
            "median_age"]
age_df.columns = new_cols
```

Employment

In [ ]:
```
emp_csv = r"C:\Users\msong\Desktop\arc2proj\data\employment\ACSST5Y2019.S2301_
data_with_overlays_2021-04-23T093758.csv"
cols =["GEO_ID",
       "S2301_C01_001E", # estimate total pop 16+
       "S2301_C02_001E" # labor force participation rate 16+
      ]
emp_df = pd.read_csv (emp_csv,
                      header=0,
                      usecols= cols,
                     )

emp_df = emp_df.iloc[1: , :] # remove first row with heading descriptions

# rename columns
new_cols = ["geo_id",
            "tot_pop_16",
            "labor_force_rate"]
emp_df.columns = new_cols
```

Household/Family

In [ ]:
```
hh_csv = r"C:\Users\msong\Desktop\arc2proj\data\householdsfam\ACSST5Y2019.S110
1_data_with_overlays_2021-04-23T093913.csv"
cols =["GEO_ID",
       "S1101_C01_001E", # total housholds
       "S1101_C01_002E", # avg household size
       "S1101_C01_003E", # total families
       "S1101_C01_004E"  # average family size
      ]
hh_df = pd.read_csv (hh_csv,
                      header=0,
                      usecols= cols,
                     )

hh_df = hh_df.iloc[1: , :] # remove first row with heading descriptions

# rename columns
new_cols = ["geo_id",
            "tot_hhs",
            "avg_hh_size",
            "tot_fams",
            "avg_fam_size"]
hh_df.columns = new_cols
```

Median Income

In [ ]:
```
inc_csv = r"C:\Users\msong\Desktop\arc2proj\data\med_income_mn\ACSST5Y2019.S19
03_data_with_overlays_2021-04-23T093941.csv"
cols =["GEO_ID",
       "S1903_C03_015E",        # med income for families households
       "S1903_C03_034E" # med income for non-family households
      ]

inc_df = pd.read_csv (inc_csv,
                      header=0,
                      usecols= cols,
                     )
inc_df = inc_df.iloc[1: , :] # remove first row with heading descriptions

# rename columns
new_cols = ["geo_id",
            "med_inc_fams",
            "med_inc_nonfams"
           ]
inc_df.columns = new_cols
```

In [ ]:
```
census_df = age_df.merge(inc_df,
                      how="left",
                      left_on="geo_id",
                      right_on="geo_id").merge(hh_df,
                                               how="left",
                                               left_on="geo_id",
                                               right_on="geo_id").merge(emp_d
f,
                                                                        how="l
eft",
                                                                        left_o
n="geo_id",
                                                                        right_
on="geo_id")
```

In [ ]:
```
# remove beginning characters of geoid
census_df['geo_id'] = census_df['geo_id'].str.replace("1400000US","").astype(s
tr)
```

In [ ]:
```
census_df =  census_df.replace("-", "0")
```

In [ ]:
```
os.chdir(r"C:\Users\msong\Desktop\arc2proj\output_data")
out_dir = os.getcwd()
census_df.to_csv(os.path.join(out_dir, "demographics.csv"), index=False)
```

In [ ]:
```
# import demographics to gdb to join with tracts
arcpy.conversion.TableToTable(r"C:\Users\msong\Desktop\arc2proj\output_data\de
mographics.csv",
                              r"C:\Users\msong\Desktop\arc2proj\Business_Fuzzy
Logic\Business_FuzzyLogic.gdb",
                              "demographics")
```

In [ ]:
```
arcpy.management.AddField("demographics",
                          "geo_id_2",
                          "TEXT")
```

In [ ]:
```
# convert geo_id to text type for joining
arcpy.management.CalculateField("demographics",
                                "geo_id_2",
                                "!geo_id!",
                                "PYTHON3")
```

In [ ]:
```
# join demographic data with metro_tracts
arcpy.management.AddJoin("metro_tracts",
                         "GEOID",
                         "demographics",
                         "geo_id_2",
                         "KEEP_COMMON")
```

In [ ]:
```

```