Data Acquisition

```
!pip install pandas matplotlib
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas)
```

Diabetic datasets Preview

```
import pandas as pd

# Load the diabetes dataset
diabetes_data = pd.read_csv('/content/Diabetes_Metrics_With_State.csv')
# Display the first few rows of the diabetes data
print("\nDiabetes Data Preview:")
print(diabetes_data.head())

# Check for general information including null values in the census,county,diabetes data
print("\nDiabetes Data Preview:")
print(diabetes_data.info())

print("\nDiabetic Missing Values in Diabetes Data:")
print(diabetes_data.isnull().sum())
```

```
Diabetes Data Preview:
   Id  Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0   1            6      148             72             35        0  33.6
1   2            1       85             66             29        0  26.6
2   3            8      183             64              0        0  23.3
3   4            1       89             66             23       94  28.1
4   5            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome         State
0                     0.627   50        1  PENNSYLVANIA
1                     0.351   31        0   PUERTO RICO
2                     0.672   32        1        NEVADA
3                     0.167   21        0       INDIANA
4                     2.288   33        1     TENNESSEE

Diabetes Data Preview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2768 entries, 0 to 2767
Data columns (total 11 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Id                        2768 non-null   int64
 1   Pregnancies               2768 non-null   int64
 2   Glucose                   2768 non-null   int64
 3   BloodPressure             2768 non-null   int64
 4   SkinThickness             2768 non-null   int64
 5   Insulin                   2768 non-null   int64
 6   BMI                       2768 non-null   float64
 7   DiabetesPedigreeFunction  2768 non-null   float64
 8   Age                       2768 non-null   int64
 9   Outcome                   2768 non-null   int64
 10  State                     2768 non-null   object
dtypes: float64(2), int64(8), object(1)
memory usage: 238.0+ KB
None

Diabetic Missing Values in Diabetes Data:
Id                          0
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
```

```
Outcome                     0
State                       0
dtype: int64
```

Census Datasets Preview

```
import pandas as pd

# Load the datasets
census_data = pd.read_csv('/content/acs2017_census_tract_data.csv')

# Display the first few rows of the census data
print("Census Data Preview:")
print(census_data.head())

# Check for general information including null values in the census,county data
print("\nCensus Data Information:")
print(census_data.info())

# Summarize missing values in both datasets
print("\nMissing Values in Census Data:")
print(census_data.isnull().sum())
```

```
Census Data Preview:
      TractId    State          County  TotalPop   Men  Women  Hispanic  \
0  1001020100  Alabama  Autauga County      1845   899    946       2.4
1  1001020200  Alabama  Autauga County      2172  1167   1005       1.1
2  1001020300  Alabama  Autauga County      3385  1533   1852       8.0
3  1001020400  Alabama  Autauga County      4267  2001   2266       9.6
4  1001020500  Alabama  Autauga County      9965  5054   4911       0.9

   White  Black  Native  ...  Walk  OtherTransp  WorkAtHome  MeanCommute  \
0   86.3    5.2     0.0  ...   0.5          0.0         2.1         24.5
1   41.6   54.5     0.0  ...   0.0          0.5         0.0         22.2
2   61.4   26.5     0.6  ...   1.0          0.8         1.5         23.1
3   80.3    7.1     0.5  ...   1.5          2.9         2.1         25.9
4   77.5   16.4     0.0  ...   0.8          0.3         0.7         21.0

   Employed  PrivateWork  PublicWork  SelfEmployed  FamilyWork  Unemployment
0       881         74.2        21.2           4.5         0.0           4.6
1       852         75.9        15.0           9.0         0.0           3.4
2      1482         73.3        21.1           4.8         0.7           4.7
3      1849         75.8        19.7           4.5         0.0           6.1
4      4787         71.4        24.1           4.5         0.0           2.3

[5 rows x 37 columns]

Census Data Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74001 entries, 0 to 74000
Data columns (total 37 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   TractId           74001 non-null  int64
 1   State             74001 non-null  object
 2   County            74001 non-null  object
 3   TotalPop          74001 non-null  int64
 4   Men               74001 non-null  int64
 5   Women             74001 non-null  int64
 6   Hispanic          73305 non-null  float64
 7   White             73305 non-null  float64
 8   Black             73305 non-null  float64
 9   Native            73305 non-null  float64
 10  Asian             73305 non-null  float64
 11  Pacific           73305 non-null  float64
 12  VotingAgeCitizen  74001 non-null  int64
 13  Income            72885 non-null  float64
 14  IncomeErr         72885 non-null  float64
 15  IncomePerCap      73256 non-null  float64
 16  IncomePerCapErr   73256 non-null  float64
 17  Poverty           73159 non-null  float64
 18  ChildPoverty      72891 non-null  float64
 19  Professional      73190 non-null  float64
 20  Service           73190 non-null  float64
 21  Office            73190 non-null  float64
 22  Construction      73190 non-null  float64
 23  Production        73190 non-null  float64
 24  Drive             73200 non-null  float64
 25  Carpool           73200 non-null  float64
 26  Transit           73200 non-null  float64
 27  Walk              73200 non-null  float64
```

Data Preprocessing

```
# Strip whitespace and convert to uppercase for merging
census_data['State'] = census_data['State'].str.strip().str.upper()
diabetes_data['State'] = diabetes_data['State'].str.strip().str.upper()
```

Check for Missing Values

```
# Fill missing values with median only for numeric columns
numeric_cols = census_data.select_dtypes(include='number').columns
census_data[numeric_cols] = census_data[numeric_cols].fillna(census_data[numeric_cols].median())
```

Aggregate Census Data by State

```
census_state_agg = census_data.groupby('State').median(numeric_only=True).reset_index()
```

Merge on State

```
merged_data = pd.merge(diabetes_data, census_state_agg, on='State', how='inner')
```

Drop Duplicates or Unused Columns

```
columns_of_interest = ['State', 'Glucose', 'BloodPressure', 'BMI', 'Age', 'Outcome',
                       'Income', 'Poverty', 'MeanCommute', 'Employed', 'Unemployment']
merged_data = merged_data[columns_of_interest]
```

```
print(merged_data.isnull().sum())
print(merged_data.describe())
```

```
State            0
Glucose          0
BloodPressure    0
BMI              0
Age              0
Outcome          0
Income           0
Poverty          0
MeanCommute      0
Employed         0
Unemployment     0
dtype: int64
            Glucose  BloodPressure          BMI          Age      Outcome  \
count  2768.000000    2768.000000  2768.000000  2768.000000  2768.000000
mean    121.102601      69.134393    32.137392    33.132225     0.343931
std      32.036508      19.231438     8.076127    11.777230     0.475104
min       0.000000       0.000000     0.000000    21.000000     0.000000
25%      99.000000      62.000000    27.300000    24.000000     0.000000
50%     117.000000      72.000000    32.200000    29.000000     0.000000
75%     141.000000      80.000000    36.625000    40.000000     1.000000
max     199.000000     122.000000    80.600000    81.000000     1.000000

              Income      Poverty  MeanCommute     Employed  Unemployment
count   2768.000000  2768.000000  2768.000000  2768.000000   2768.000000
mean   55396.956105    13.080076    23.855853  1852.321893      5.816944
std    11093.758862     5.357183     3.838887   261.068775      1.895079
min    19132.000000     6.700000    16.800000  1016.000000      2.300000
25%    48125.000000     9.900000    21.300000  1656.500000      4.800000
50%    54375.000000    12.100000    23.800000  1859.000000      5.900000
75%    61611.000000    14.700000    25.900000  2069.000000      6.500000
max    79306.000000    45.100000    33.200000  2364.000000     16.400000
```

Data Merging

```
import pandas as pd

# Load datasets
diabetes_data = pd.read_csv('/content/Diabetes_Metrics_With_State.csv')
census_data = pd.read_csv('/content/acs2017_census_tract_data.csv')

# Convert 'State' columns to uppercase to ensure matching
diabetes_data['State'] = diabetes_data['State'].str.upper()
census_data['State'] = census_data['State'].str.upper()

# Aggregate census data at the state level (mean values)
census_state_summary = census_data.groupby('State')[['Income', 'Poverty', 'MeanCommute', 'Employed', 'Unemployment']].mean().re

# Merge diabetes and summarized census data on State
merged_data = pd.merge(diabetes_data, census_state_summary, on='State', how='inner')
```

```
# Select final columns
selected_columns = [
    'State', 'Glucose', 'BloodPressure', 'BMI', 'Age', 'Outcome',
    'Income', 'Poverty', 'MeanCommute', 'Employed', 'Unemployment'
]

viz_data = merged_data[selected_columns]

# Save to CSV
viz_data.to_csv('/content/merged_dataset.csv', index=False)

print("✅ Visualization-ready dataset created successfully!")
print(viz_data.head())
```

```
✅ Visualization-ready dataset created successfully!
            State  Glucose  BloodPressure   BMI  Age  Outcome        Income  \
0  PENNSYLVANIA      148             72  33.6   50        1  59459.747643
1   PUERTO RICO       85             66  26.6   31        0  21206.667429
2        NEVADA      183             64  23.3   32        1  58820.623894
3       INDIANA       89             66  28.1   21        0  51832.129333
4     TENNESSEE      137             40  43.1   33        1  50463.940857

      Poverty  MeanCommute     Employed  Unemployment
0  14.420763    26.470801  1894.647918      7.044292
1  46.247458    28.281087  1100.673016     19.011964
2  14.627434    23.829056  1952.486172      8.348525
3  16.995216    23.149035  2067.700199      6.903254
4  18.219499    24.576626  2001.743487      7.320365
```