Demand-Side Data Collection and Processing

The demand-side portion of the childcare supply–demand model for **Howard County, Maryland** is constructed from two primary components:

1. **Demographics and child population estimates**
   (ACS 2018–2022 5-year estimates)
2. **Employment and commuter flows**
   (LEHD LODES8 Origin–Destination, 2022)

These two pieces together allow us to estimate where children under age 5 live, how many children are likely to need care during parental working hours, and how daytime population shifts based on commuter activity.

1.1. Demographics Table Construction (demographics_block)

1.1.1 Data Sources

The notebook *Howard_County_demographics_block.ipynb* extracts demographic data from:

- **ACS 5-year 2018–2022**
- **Table B09001** — *Children by age*: used to compute counts of
  - Children aged **0–2**
  - Children aged **3–5**
  - **Total under age 5**
- **Table B19013** — *Median Household Income*
- **Table C17002** — *Ratio of Income to Poverty Level (Categories)*
- **ACS PUMS (optional)** — used to refine the 0–2 vs. 3–5 age split
- **TIGER/Line geographic identifiers** for Census tracts and block groups

All processing is restricted to **Howard County (FIPS: 24027)**.

1.1.2 Overview of Python Workflow

The notebook performs four major steps:

Step A — Extract Under-5 Counts from ACS B09001

The ACS table **B09001** provides child counts in fine-grained age bins.

Procedure:

1. Data is downloaded via the Census API using a function such as:

```
def acs_blockgroup(year, table, state, county):
    url = f".../acs/5yr/{year}/.../{table}..."
    # returns block-group level ACS results as pandas DataFrame
```

2. The notebook selects the bins representing:
    - **0–2 years**
    - **3–5 years**

Depending on the available B09001 bins for block groups, the notebook then:

- **Directly uses age bins** if provided, OR
- **Uses PUMS state-level proportions** to split “under 5” into 0–2 vs. 3–5 categories.

PUMS-Based Age Split (Shown in Notebook)

Because ACS block group does not provide exact 0–2 vs. 3–5 splits, the notebook compute **state-level shares** from Maryland PUMS:

```
share_0_2 = (children_age_0_2_in_PUMS / total_children_under5_PUMS)
share_3_5 = (children_age_3_5_in_PUMS / total_children_under5_PUMS)
```

Then applies these shares proportionally within each block group:

```
bg["children_0_2"] = bg["children_under5_total"] * share_0_2
bg["children_3_5"] = bg["children_under5_total"] * share_3_5
```

This preserves total under-5 counts while allocating them realistically across age groups.

Step B — Add Median Household Income (ACS B19013)

The notebook loads **tract-level** median household income from B19013:

```
tract_income = acs_tract("B19013", state="24", county="027")
```

Because income is not available at the block-group level, the notebook:

Apportions tract-level income to block groups

based on each block group's share of under-5 children within its tract:

$$Income_{bg} = Income_{tract} \times \left(\frac{\text{children\_under5}_{bg}}{\sum_{bg \in tract} \text{children\_under5}} \right)$$

This ensures demographic-weighted allocation aligned with childcare demand.

Step C — Compute Poverty Level Ratio (ACS C17002)

ACS table **C17002** provides population counts in income-to-poverty categories:

Example categories:

- < 0.50 FPL
- 0.50–0.99 FPL
- 1.00–1.99 FPL
- ≥ 2.00 FPL

Notebook Formula:

1. Assign each category a **midpoint ratio**
   (e.g., <0.50 → 0.25, 0.50–0.99 → 0.75, 1.00–1.99 → 1.50, ≥2.00 → 2.50)

2. Compute weighted average:

$$poverty\_level\_ratio = \frac{\sum(\text{midpoint}_c \times population_c)}{\text{total population}}$$

This provides a continuous poverty metric:

- **< 1.0** → average household below Federal Poverty Line
- **1–2** → low-income
- **> 2** → moderate/high income

This is crucial for later estimating **subsidy-eligible childcare demand**.

Step D — Final Block Group Table Assembly

After merging all above pieces, the notebook outputs:

Fields produced:

Column	**Description**
geo_id	Block group GEOID
children_0_2	Estimated children age 0–2
children_3_5	Estimated children age 3–5
children_total_under5	Sum of above
median_household_income	Apportioned tract-level income
poverty_level_ratio	Weighted average ratio
data_year	2022
source	ACS 5-year 2018–2022

1.2. Employee Commute Table Construction (employee_commute)

This section is built from the notebook **Howard_County_employee_commute.ipynb**, which processes **LEHD LODES8 OD 2022** commuter data.

3.2.1 Data Sources

- **LODES8 Origin-Destination (OD)**
  Files used:

  ```
  md_od_main_JTxx_2022.csv.gz
  ```

  ```
  md_od_aux_JTxx_2022.csv.gz
  ```

Contains:

- h_geocode — home census block (15-digit)
- w_geocode — work census block (15-digit)
- Six job types (JT00–JT05)
- Column S000 — total jobs (all workers)

The notebook merges all job types and both OD parts into one combined dataset.

3.2.2 Python Workflow

Step A — Load and Merge All LODES Files

The notebook loops over:

```
PARTS = ["main", "aux"]

JOB_TYPES = ["JT00","JT01","JT02","JT03","JT04","JT05"]
```

and reads every compressed file:

```
with gzip.open(path) as f:

    chunk = pd.read_csv(f, header=None, names=LODES_COLS)
```

All files are concatenated into one master dataframe od.

Step B — Filter to Howard County OD Flows

Howard County FIPS = **24027**
Notebook filters OD pairs where **either** home or work block belongs to the county:

```
pair = od[(od["h_geocode"].str.startswith("24027")) |
```

```
    (od["w_geocode"].str.startswith("24027"))]
```

This keeps all commuting relationships relevant to the county.

Step C — Compute Net Commute Flow per Block

The notebook computes **inflow**, **outflow**, and **net flow** for each block.

Definitions:

$$outflow(b) = \sum_{home=b, work \neq b} S000$$

$$inflow(b) = \sum_{work=b, home \neq b} S000$$

$$net\_commute\_flow(b) = inflow(b) - outflow(b)$$

Notebook Implementation:

```
outflow = pair.groupby("h_geocode")["S000"].sum()

inflow  = pair.groupby("w_geocode")["S000"].sum()

net = pd.DataFrame({

    "outflow_bg": outflow,

    "inflow_bg": inflow

}).fillna(0)

net["net_commute_flow_bg"] = net["inflow_bg"] - net["outflow_bg"]
```

Only blocks inside Howard County (str.slice(0,5) == "24027") are kept.

Step D — Attach Net Flow Back to Each OD Pair

Each OD pair receives a net_commute_flow based on:

- If **work block** is in Howard → use its net flow
- Else if **home block** is in Howard → use its net flow
- Else → 0

Notebook code:

```
pair = pair.merge(net_howard.rename(...), on="work_geo_id", how="left")

pair = pair.merge(net_howard.rename(...), on="home_geo_id", how="left")

def choose_net(row):

    if notna(row["_net_work"]): return row["_net_work"]

    if notna(row["_net_home"]): return row["_net_home"]

    return 0

pair["net_commute_flow"] = pair.apply(choose_net, axis=1)
```

This ensures every Howard-related OD pair has an associated net flow measure.

Step E — Final Output

The final table contains:

Column	Description
home_geo_id	Home census block
work_geo_id	Work census block
commuter_count / S000	Worker flow volume
net_commute_flow	Net inflow/outflow of chosen block
data_year	2022
source	LODES8

3.3. Relationship Between Demographics and Commute Data

The two datasets work together:

Demographics (ACS) provides:

- Where children under 5 live
- Income & poverty indicators

- Disadvantaged or high-need areas

Commute flows (LODES) provide:

- Where parents *work*
- Daytime population shifts
- Identification of job centers with strong worker inflow

This allows the model to produce:

- **Residence-based childcare demand**
- **Workplace-adjusted childcare demand**
- **Total demand = max(residential, workplace-shifted)**

Later combined with the supply dataset to compute:

- **Absolute childcare shortage**
- **Supply–demand ratio**
- **Relative shortage score**

Methods Summary (Demand-Side Data Collection)

Demand-side childcare estimates for Howard County were generated using two complementary data pipelines: (1) **ACS-based demographic extraction**, and (2) **LODES-based commuter flow modeling**. Together, these datasets provide a detailed picture of where young children live, how many are likely to need childcare, and how daytime population shifts based on parental work patterns.

First, demographic data at the census block group level were extracted using **ACS 2018–2022 5-year estimates**. Table **B09001** was used to identify the total number of children under age five, and a proportional age-split between ages 0–2 and 3–5 was calculated using **Maryland PUMS** microdata. Median household income was obtained from **Table B19013** and apportioned from tract to block group based on each block group's share of under-5 children in its tract. A continuous poverty indicator was created from **Table C17002**, which categorizes population counts by income-to-poverty level ratios. Category midpoints were used to compute a weighted average poverty ratio for each block group, producing a smooth measure of socioeconomic vulnerability.

Second, employment-related childcare demand was modeled using **LEHD LODES8 Origin–Destination (OD) 2022** data. All OD flows involving Howard County were extracted, and total worker counts (S000) were aggregated. For each census block, the number of workers commuting in (inflow) and out (outflow) was computed, and a **net commute flow** metric was derived. Positive values indicate job centers, while negative values indicate primarily residential blocks. Each OD record was then assigned the appropriate net flow value depending on whether the home or work block belonged to Howard County.

Together, the **demographics_block** and **employee_commute** datasets form the foundation of the childcare demand model, supporting both residence-based and workplace-shift–adjusted demand calculations.