02-gentrification-early-warning

November 29, 2025

1 Gentrification Early Warning System

Detecting Neighborhood Change Through Demographic and Economic Indicators

---

1.1 Executive Summary

This notebook demonstrates how to use the **KRL Suite** to build a gentrification early warning system using real Census ACS demographic data combined with economic indicators from FRED and BLS.

1.1.1 KRL Suite Components Used

- **krl_data_connectors.community**: `CensusACSPublicConnector` for state-level demographics
- **krl_data_connectors.community**: `FREDBasicConnector`, `BLSBasicConnector` for economic context
- **krl_core**: Logging and utilities

1.1.2 Key Indicators for Gentrification Risk

1. **Demographic shifts**: Changes in income, education, race/ethnicity
2. **Economic pressure**: Wage growth vs inflation (from previous notebook)
3. **Housing market stress**: Mortgage rates and construction activity

1.1.3 What You'll Learn

1. Fetching state-level demographic data from Census ACS
2. Calculating gentrification risk indicators
3. Comparing demographic profiles across states
4. Identifying areas with high change velocity

Estimated Time: 20-25 minutes
Difficulty: Intermediate

> **Note:** Community tier provides state-level data. Professional tier unlocks county/tract-level analysis.

1.2 Table of Contents

1. Setup and Imports

[3]:
```
# Standard library imports
import os
import sys
import warnings
from datetime import datetime, timedelta
import importlib

# Add KRL package paths (handles spaces in path correctly)
_krl_base = os.path.expanduser("~/Documents/GitHub/KRL/Private IP")
for _pkg in ["krl-open-core/src", "krl-data-connectors/src"]:
    _path = os.path.join(_krl_base, _pkg)
    if _path not in sys.path:
        sys.path.insert(0, _path)

# Load environment variables from .env file
from dotenv import load_dotenv
_env_path = os.path.expanduser("~/Documents/GitHub/KRL/krl-tutorials/.env")
load_dotenv(_env_path)

# Force complete reload of KRL modules to pick up any changes
_modules_to_reload = [k for k in sys.modules.keys() if k.
 ↪startswith(('krl_core', 'krl_data_connectors'))]
for _mod in _modules_to_reload:
    del sys.modules[_mod]

# Data manipulation
import pandas as pd
import numpy as np
from scipy import stats

# Visualization
import plotly.express as px
```

```python
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# =============================================================================
# KRL Suite Imports - REAL package imports
# =============================================================================

# KRL Data Connectors - Community Tier (Free, no API key required)
from krl_data_connectors.community import (
    CensusACSPublicConnector,  # Census American Community Survey
    FREDBasicConnector,         # Federal Reserve Economic Data
    BLSBasicConnector,          # Bureau of Labor Statistics
)

# KRL Core - Logging utilities
from krl_core import get_logger

# Configure display
pd.set_option('display.max_columns', 25)
pd.set_option('display.float_format', '{:,.2f}'.format)
warnings.filterwarnings('ignore', category=FutureWarning)

# Initialize logger
logger = get_logger("GentrificationEarlyWarning")

print("=" * 65)
print("  Gentrification Early Warning System")
print("=" * 65)
print(f"  Execution Time: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}")
print(f"  Using KRL Data Connectors (Community Tier)")
print(f"  FRED API Key: {'  Loaded' if os.getenv('FRED_API_KEY') else '  Not␣
↪found'}")
print("=" * 65)
```

```
=================================================================
 Gentrification Early Warning System
=================================================================
 Execution Time: 2025-11-27 11:59:03
 Using KRL Data Connectors (Community Tier)
 FRED API Key:   Loaded
=================================================================
```

2. Data Loading: Census ACS Demographics

We'll use the `CensusACSPublicConnector` to fetch state-level demographic data that can indicate gentrification pressures:

Variable	Code	Description
Population	B01001_001E	Total population
Median Income	B19013_001E	Median household income
Poverty	B17001_002E	Population below poverty level
Education	B15003_022E, B15003_023E	Bachelor's/Master's degree holders
Housing	B25077_001E	Median home value

Community Tier: State-level data only. For tract-level gentrification detection, upgrade to Professional.

[4]:
```python
# ============================================================================
# Initialize KRL Data Connectors
# ============================================================================

# Initialize Census ACS connector
census = CensusACSPublicConnector()

# Initialize FRED connector for economic context
fred = FREDBasicConnector()

# Test connections
print("  Testing API Connections...")
print(f"   Census API Connected: {census.connect()}")
print(f"   FRED API Connected: {fred.connect()}")

# List available Census variables
print("\n  Available Census ACS Variables (Community Tier):")
for code, desc in list(census.COMMON_VARIABLES.items())[:8]:
    print(f"   • {code}: {desc}")
print("   ...")
```

{"timestamp": "2025-11-27T16:59:11.641512Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Connector initialized", "source": {"file": "base_connector.py", "line": 81, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "connector": "CensusACSPublicConnector", "cache_dir": "/Users/bcdelo/.krl_cache/censusacspublicconnector", "cache_ttl": 3600, "has_api_key": true}
{"timestamp": "2025-11-27T16:59:11.641874Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Initialized Census ACS Public connector (Community tier)", "source": {"file": "census_acs_public.py", "line": 101, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "geography": "state-level only"}
{"timestamp": "2025-11-27T16:59:11.642873Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Connector initialized", "source": {"file": "base_connector.py", "line": 81, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "connector": "FREDBasicConnector", "cache_dir": "/Users/bcdelo/.krl_cache/fredbasicconnector", "cache_ttl": 3600, "has_api_key":

true}
{"timestamp": "2025-11-27T16:59:11.643169Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Initialized FRED Basic connector (Community tier)", "source": {"file": "fred_basic.py", "line": 96, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "available_series": 15}
{"timestamp": "2025-11-27T16:59:11.641874Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Initialized Census ACS Public connector (Community tier)", "source": {"file": "census_acs_public.py", "line": 101, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "geography": "state-level only"}
{"timestamp": "2025-11-27T16:59:11.642873Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Connector initialized", "source": {"file": "base_connector.py", "line": 81, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "connector": "FREDBasicConnector", "cache_dir": "/Users/bcdelo/.krl_cache/fredbasicconnector", "cache_ttl": 3600, "has_api_key": true}
{"timestamp": "2025-11-27T16:59:11.643169Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Initialized FRED Basic connector (Community tier)", "source": {"file": "fred_basic.py", "line": 96, "function": "__init__"}, "levelname": "INFO", "taskName": "Task-39", "available_series": 15}
Testing API Connections…
Testing API Connections…
{"timestamp": "2025-11-27T16:59:12.143376Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Successfully connected to Census API", "source": {"file": "census_acs_public.py", "line": 137, "function": "connect"}, "levelname": "INFO", "taskName": "Task-39"}
{"timestamp": "2025-11-27T16:59:12.143376Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Successfully connected to Census API", "source": {"file": "census_acs_public.py", "line": 137, "function": "connect"}, "levelname": "INFO", "taskName": "Task-39"}
Census API Connected: True
Census API Connected: True
{"timestamp": "2025-11-27T16:59:12.418195Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Successfully connected to FRED API", "source": {"file": "fred_basic.py", "line": 131, "function": "connect"}, "levelname": "INFO", "taskName": "Task-39"}
FRED API Connected: True

Available Census ACS Variables (Community Tier):
- B01001_001E: Total Population
- B01002_001E: Median Age
- B01001_002E: Male Population
- B01001_026E: Female Population
- B02001_002E: White Alone
- B02001_003E: Black or African American Alone
- B02001_005E: Asian Alone
- B03003_003E: Hispanic or Latino

…

{"timestamp": "2025-11-27T16:59:12.418195Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Successfully connected to FRED API", "source": {"file": "fred_basic.py", "line": 131, "function": "connect"}, "levelname": "INFO", "taskName": "Task-39"}
FRED API Connected: True

Available Census ACS Variables (Community Tier):

- B01001_001E: Total Population
- B01002_001E: Median Age
- B01001_002E: Male Population
- B01001_026E: Female Population
- B02001_002E: White Alone
- B02001_003E: Black or African American Alone
- B02001_005E: Asian Alone
- B03003_003E: Hispanic or Latino

…

[5]:

```python
# ============================================================================
# Fetch State-Level Demographics (2022 - most recent 5-year estimates)
# ============================================================================

# Get comprehensive demographics by state
demographics_2022 = census.get_demographics_by_state(year=2022)

print("  Demographics Data Retrieved:")
print(f"   States: {len(demographics_2022)}")
print(f"   Variables: {list(demographics_2022.columns)}")
demographics_2022.head(10)
```

{"timestamp": "2025-11-27T16:59:18.929667Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Fetching Census ACS data for 2022", "source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-42", "year": 2022, "variables": 9, "geography": "state"}
{"timestamp": "2025-11-27T16:59:19.547539Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source": {"file": "census_acs_public.py", "line": 197, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-42", "year": 2022, "rows": 52}
Demographics Data Retrieved:
States: 52
Variables: ['NAME', 'B01001_001E', 'B01002_001E', 'B02001_002E', 'B02001_003E', 'B02001_005E', 'B03003_003E', 'B19013_001E', 'B17001_002E', 'state']
{"timestamp": "2025-11-27T16:59:19.547539Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source": {"file": "census_acs_public.py", "line": 197, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-42", "year": 2022, "rows": 52}
Demographics Data Retrieved:

```
States: 52
Variables: ['NAME', 'B01001_001E', 'B01002_001E', 'B02001_002E',
'B02001_003E', 'B02001_005E', 'B03003_003E', 'B19013_001E', 'B17001_002E',
'state']
```

[5]:

```
                   NAME  B01001_001E  B01002_001E  B02001_002E  B02001_003E  \
0               Alabama      5028092        39.30      3329012      1326341
1                Alaska       734821        35.30       450472        23395
2               Arizona      7172282        38.40      4781702       327077
3              Arkansas      3018669        38.40      2193348       456693
4            California     39356104        37.30     18943660      2202587
5              Colorado      5770790        37.30      4393409       233712
6           Connecticut      3611317        40.90      2522166       385407
7              Delaware       993635        41.40       634244       218266
8  District of Columbia       670587        34.80       265633       297101
9               Florida     21634529        42.40     13807410      3355708

   B02001_005E  B03003_003E  B19013_001E  B17001_002E state
0        69808       232407        59609       768897    01
1        47464        54890        86370        75227    02
2       240642      2297513        72581       916876    04
3        47413       243321        56335       475729    05
4      5949136     15617930        91905      4685272    06
5       185431      1273762        87598       540105    08
6       170945       627408        90213       355692    09
7        40570        98696        79325       107790    10
8        27067        77168       101722        98039    11
9       609990      5738283        67917      2725633    12
```

[6]:

```python
# ============================================================================
# Fetch Multi-Year Data to Detect Change Velocity
# ============================================================================

# Get demographics for multiple years to calculate change rates
years = [2017, 2019, 2021, 2022]
multi_year_data = {}

for year in years:
    try:
        df = census.get_demographics_by_state(year=year)
        df['year'] = year
        multi_year_data[year] = df
        print(f"    {year}: {len(df)} states")
    except Exception as e:
        print(f"    {year}: Error - {str(e)[:50]}")

# Combine all years
```

```python
if multi_year_data:
    all_demographics = pd.concat(multi_year_data.values(), ignore_index=True)
    print(f"\n Combined Dataset: {len(all_demographics)} rows x {len(all_demographics.columns)} columns")
```

{"timestamp": "2025-11-27T16:59:25.224027Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Fetching Census ACS data for 2017", "source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2017, "variables": 9, "geography": "state"}
{"timestamp": "2025-11-27T16:59:25.810848Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source": {"file": "census_acs_public.py", "line": 197, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2017, "rows": 52}
    2017: 52 states
{"timestamp": "2025-11-27T16:59:25.811944Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Fetching Census ACS data for 2019", "source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2019, "variables": 9, "geography": "state"}
{"timestamp": "2025-11-27T16:59:25.810848Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source": {"file": "census_acs_public.py", "line": 197, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2017, "rows": 52}
    2017: 52 states
{"timestamp": "2025-11-27T16:59:25.811944Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Fetching Census ACS data for 2019", "source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2019, "variables": 9, "geography": "state"}
{"timestamp": "2025-11-27T16:59:26.408192Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source": {"file": "census_acs_public.py", "line": 197, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2019, "rows": 52}
    2019: 52 states
{"timestamp": "2025-11-27T16:59:26.409616Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Fetching Census ACS data for 2021", "source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2021, "variables": 9, "geography": "state"}
{"timestamp": "2025-11-27T16:59:26.408192Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source": {"file": "census_acs_public.py", "line": 197, "function": "get_data"}, "levelname": "INFO", "taskName": "Task-45", "year": 2019, "rows": 52}
    2019: 52 states
{"timestamp": "2025-11-27T16:59:26.409616Z", "level": "INFO", "name": "CensusACSPublicConnector", "message": "Fetching Census ACS data for 2021", "source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"},

"levelname": "INFO", "taskName": "Task-45", "year": 2021, "variables": 9,
"geography": "state"}
{"timestamp": "2025-11-27T16:59:26.938431Z", "level": "INFO", "name":
"CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source":
{"file": "census_acs_public.py", "line": 197, "function": "get_data"},
"levelname": "INFO", "taskName": "Task-45", "year": 2021, "rows": 52}
```
    2021: 52 states
```
{"timestamp": "2025-11-27T16:59:26.939030Z", "level": "INFO", "name":
"CensusACSPublicConnector", "message": "Fetching Census ACS data for 2022",
"source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"},
"levelname": "INFO", "taskName": "Task-45", "year": 2022, "variables": 9,
"geography": "state"}
{"timestamp": "2025-11-27T16:59:26.938431Z", "level": "INFO", "name":
"CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source":
{"file": "census_acs_public.py", "line": 197, "function": "get_data"},
"levelname": "INFO", "taskName": "Task-45", "year": 2021, "rows": 52}
```
    2021: 52 states
```
{"timestamp": "2025-11-27T16:59:26.939030Z", "level": "INFO", "name":
"CensusACSPublicConnector", "message": "Fetching Census ACS data for 2022",
"source": {"file": "census_acs_public.py", "line": 175, "function": "get_data"},
"levelname": "INFO", "taskName": "Task-45", "year": 2022, "variables": 9,
"geography": "state"}
{"timestamp": "2025-11-27T16:59:27.225253Z", "level": "INFO", "name":
"CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source":
{"file": "census_acs_public.py", "line": 197, "function": "get_data"},
"levelname": "INFO", "taskName": "Task-45", "year": 2022, "rows": 52}
```
    2022: 52 states

  Combined Dataset: 208 rows x 11 columns
```
{"timestamp": "2025-11-27T16:59:27.225253Z", "level": "INFO", "name":
"CensusACSPublicConnector", "message": "Retrieved data for 52 states", "source":
{"file": "census_acs_public.py", "line": 197, "function": "get_data"},
"levelname": "INFO", "taskName": "Task-45", "year": 2022, "rows": 52}
```
    2022: 52 states

  Combined Dataset: 208 rows x 11 columns
```

3. Signal Processing: Income & Education Changes

Gentrification typically manifests through: - **Rising median incomes** (higher-earning residents moving in) - **Increasing education levels** (more college graduates) - **Shifting poverty rates** (lower poverty as demographics change)

[7]:
```
# ============================================================================
# Calculate Change Velocity (2017 to 2022)
# ============================================================================

# Pivot data to compare years
```

```
if 2017 in multi_year_data and 2022 in multi_year_data:
    df_2017 = multi_year_data[2017].set_index('NAME')
    df_2022 = multi_year_data[2022].set_index('NAME')

    # Calculate percentage changes
    change_df = pd.DataFrame({
        'state': df_2022.index,
        'population_2017': df_2017['B01001_001E'].values,
        'population_2022': df_2022['B01001_001E'].values,
        'income_2017': df_2017['B19013_001E'].values,
        'income_2022': df_2022['B19013_001E'].values,
        'poverty_2017': df_2017['B17001_002E'].values,
        'poverty_2022': df_2022['B17001_002E'].values,
    })

    # Compute growth rates
    change_df['pop_growth_pct'] = ((change_df['population_2022'] /␣
↪change_df['population_2017']) - 1) * 100
    change_df['income_growth_pct'] = ((change_df['income_2022'] /␣
↪change_df['income_2017']) - 1) * 100
    change_df['poverty_rate_2017'] = (change_df['poverty_2017'] /␣
↪change_df['population_2017']) * 100
    change_df['poverty_rate_2022'] = (change_df['poverty_2022'] /␣
↪change_df['population_2022']) * 100
    change_df['poverty_change_pts'] = change_df['poverty_rate_2022'] -␣
↪change_df['poverty_rate_2017']

    print("  Change Metrics (2017-2022):")
    change_df[['state', 'pop_growth_pct', 'income_growth_pct',␣
↪'poverty_change_pts']].head(10)
```

```
 Change Metrics (2017-2022):
```

[8]:
```
# ==============================================================================
# Identify High-Change States (Gentrification Pressure Indicators)
# ==============================================================================

# States with strong gentrification signals:
# - High income growth
# - Population growth
# - Declining poverty rates

# Compute composite score
change_df['gentrification_score'] = (
    change_df['income_growth_pct'].rank(pct=True) * 0.4 +
    change_df['pop_growth_pct'].rank(pct=True) * 0.3 +
    (-change_df['poverty_change_pts']).rank(pct=True) * 0.3
```

```
) * 100

# Rank states
change_df = change_df.sort_values('gentrification_score', ascending=False)
change_df['rank'] = range(1, len(change_df) + 1)

print("  Top 10 States by Gentrification Pressure Score:")
change_df[['state', 'income_growth_pct', 'pop_growth_pct',␣
 ↪'poverty_change_pts', 'gentrification_score', 'rank']].head(10)
```

```
 Top 10 States by Gentrification Pressure Score:
```

[8]:

	state	income_growth_pct	pop_growth_pct	poverty_change_pts	gentrification_score	rank
32	New York	311.56	476.38	-31.20	94.23	1
29	New Hampshire	106.18	-24.89	-10.15	81.54	2
21	Massachusetts	97.83	42.73	-6.53	81.35	3
44	Utah	86.60	-25.78	-9.33	78.85	4
4	California	65.79	1,262.88	-2.13	77.88	5
20	Maryland	61.30	483.42	-3.82	77.50	6
23	Minnesota	73.10	-13.67	-7.19	77.50	7
11	Hawaii	116.41	-51.29	-8.23	76.54	8
46	Virginia	64.55	548.38	-2.85	75.38	9
38	Pennsylvania	43.51	683.72	-2.83	72.88	10

4. Economic Context: Housing & Labor Market

Let's add economic context from FRED to understand the broader pressures driving gentrification.

[9]:
```
# ============================================================================
# Fetch Economic Context from FRED
# ============================================================================

# Housing starts - indicates construction activity
housing_starts = fred.get_series("HOUST", start_date="2017-01-01",␣
 ↪end_date="2022-12-31")

# Mortgage rates - affects affordability pressure
```

```python
mortgage_rates = fred.get_series("MORTGAGE30US", start_date="2017-01-01", end_date="2022-12-31")

# CPI - general inflation pressure
cpi = fred.get_series("CPIAUCSL", start_date="2017-01-01", end_date="2022-12-31")

print("  Economic Context Data Retrieved:")
print(f"   Housing Starts: {len(housing_starts)} observations")
print(f"   Mortgage Rates: {len(mortgage_rates)} observations")
print(f"   CPI: {len(cpi)} observations")

# Calculate summary stats
print(f"\n  Key Economic Trends (2017-2022):")
print(f"   Housing Starts (avg): {housing_starts['value'].mean():.0f}k units/month")
print(f"   Mortgage Rate (start): {mortgage_rates['value'].iloc[0]:.2f}%")
print(f"   Mortgage Rate (end): {mortgage_rates['value'].iloc[-1]:.2f}%")
print(f"   CPI Growth: {((cpi['value'].iloc[-1] / cpi['value'].iloc[0]) - 1) * 100:.1f}%")
```

{"timestamp": "2025-11-27T16:59:46.499487Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Fetching FRED series: HOUST", "source": {"file": "fred_basic.py", "line": 167, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "HOUST", "start_date": "2017-01-01", "end_date": "2022-12-31"}
{"timestamp": "2025-11-27T16:59:46.601409Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Retrieved 72 observations for HOUST", "source": {"file": "fred_basic.py", "line": 197, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "HOUST", "rows": 72}
{"timestamp": "2025-11-27T16:59:46.602018Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Fetching FRED series: MORTGAGE30US", "source": {"file": "fred_basic.py", "line": 167, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "MORTGAGE30US", "start_date": "2017-01-01", "end_date": "2022-12-31"}
{"timestamp": "2025-11-27T16:59:46.601409Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Retrieved 72 observations for HOUST", "source": {"file": "fred_basic.py", "line": 197, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "HOUST", "rows": 72}
{"timestamp": "2025-11-27T16:59:46.602018Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Fetching FRED series: MORTGAGE30US", "source": {"file": "fred_basic.py", "line": 167, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "MORTGAGE30US", "start_date": "2017-01-01", "end_date": "2022-12-31"}
{"timestamp": "2025-11-27T16:59:46.705248Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Retrieved 313 observations for MORTGAGE30US", "source": {"file": "fred_basic.py", "line": 197, "function": "get_series"},

"levelname": "INFO", "taskName": "Task-75", "series_id": "MORTGAGE30US", "rows": 313}
{"timestamp": "2025-11-27T16:59:46.705885Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Fetching FRED series: CPIAUCSL", "source": {"file": "fred_basic.py", "line": 167, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "CPIAUCSL", "start_date": "2017-01-01", "end_date": "2022-12-31"}
{"timestamp": "2025-11-27T16:59:46.705248Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Retrieved 313 observations for MORTGAGE30US", "source": {"file": "fred_basic.py", "line": 197, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "MORTGAGE30US", "rows": 313}
{"timestamp": "2025-11-27T16:59:46.705885Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Fetching FRED series: CPIAUCSL", "source": {"file": "fred_basic.py", "line": 167, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "CPIAUCSL", "start_date": "2017-01-01", "end_date": "2022-12-31"}
{"timestamp": "2025-11-27T16:59:46.825083Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Retrieved 72 observations for CPIAUCSL", "source": {"file": "fred_basic.py", "line": 197, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "CPIAUCSL", "rows": 72}

```
 Economic Context Data Retrieved:
   Housing Starts: 72 observations
   Mortgage Rates: 313 observations
   CPI: 72 observations

 Key Economic Trends (2017-2022):
   Housing Starts (avg): 1382k units/month
   Mortgage Rate (start): 4.20%
   Mortgage Rate (end): 6.42%
   CPI Growth: 22.7%
```

{"timestamp": "2025-11-27T16:59:46.825083Z", "level": "INFO", "name": "FREDBasicConnector", "message": "Retrieved 72 observations for CPIAUCSL", "source": {"file": "fred_basic.py", "line": 197, "function": "get_series"}, "levelname": "INFO", "taskName": "Task-75", "series_id": "CPIAUCSL", "rows": 72}

```
 Economic Context Data Retrieved:
   Housing Starts: 72 observations
   Mortgage Rates: 313 observations
   CPI: 72 observations

 Key Economic Trends (2017-2022):
   Housing Starts (avg): 1382k units/month
   Mortgage Rate (start): 4.20%
   Mortgage Rate (end): 6.42%
   CPI Growth: 22.7%
```

[10]:
```python
# ============================================================================
# Compare Economic Pressure to Demographic Change
# ============================================================================

# Get annual averages for economic indicators
housing_annual = housing_starts.resample('YS').mean()
mortgage_annual = mortgage_rates.resample('YS').mean()

economic_context = pd.DataFrame({
    'year': housing_annual.index.year,
    'housing_starts_avg': housing_annual['value'].values,
    'mortgage_rate_avg': mortgage_annual['value'].values,
})
economic_context = economic_context.set_index('year')

print("  Annual Economic Context:")
economic_context
```

```
 Annual Economic Context:
```

[10]:

year	housing_starts_avg	mortgage_rate_avg
2017	1,204.67	3.99
2018	1,246.83	4.54
2019	1,291.50	3.94
2020	1,394.33	3.11
2021	1,603.17	2.96
2022	1,551.50	5.34

5. Visualization: Gentrification Pressure Analysis

Let's visualize the demographic changes and identify patterns.

[11]:
```python
# ============================================================================
# Visualization 1: Income Growth vs Population Growth
# ============================================================================

fig = px.scatter(
    change_df,
    x='pop_growth_pct',
    y='income_growth_pct',
    color='gentrification_score',
    size=abs(change_df['poverty_change_pts']) + 1,
    hover_name='state',
    title='State Demographic Changes: Income vs Population Growth (2017-2022)',
    labels={
        'pop_growth_pct': 'Population Growth (%)',
        'income_growth_pct': 'Median Income Growth (%)',
        'gentrification_score': 'Gentrification Score'
```

```
    },
    color_continuous_scale='RdYlGn_r',
    template='plotly_white',
)

# Add quadrant lines
fig.add_vline(x=change_df['pop_growth_pct'].median(), line_dash="dash",␣
 ↪line_color="gray", opacity=0.5)
fig.add_hline(y=change_df['income_growth_pct'].median(), line_dash="dash",␣
 ↪line_color="gray", opacity=0.5)

fig.update_layout(height=550)
fig.show()
```

[12]:
```
# ==============================================================================
# Visualization 2: Top 15 States by Gentrification Pressure
# ==============================================================================

top_15 = change_df.head(15).copy()

fig = go.Figure()

# Income growth bars
fig.add_trace(go.Bar(
    name='Income Growth %',
    x=top_15['state'],
    y=top_15['income_growth_pct'],
    marker_color='#0077BB',
))

# Population growth bars
fig.add_trace(go.Bar(
    name='Population Growth %',
    x=top_15['state'],
    y=top_15['pop_growth_pct'],
    marker_color='#009988',
))

# Poverty change (inverted - negative is good for gentrification signal)
fig.add_trace(go.Bar(
    name='Poverty Rate Change (pts)',
    x=top_15['state'],
    y=top_15['poverty_change_pts'],
    marker_color='#EE7733',
))

fig.update_layout(
```

```
    title='Top 15 States: Gentrification Pressure Indicators (2017-2022)',
    xaxis_title='State',
    yaxis_title='Percentage Change',
    barmode='group',
    template='plotly_white',
    xaxis_tickangle=-45,
    height=500,
    legend=dict(yanchor="top", y=0.99, xanchor="right", x=0.99),
)

fig.show()
```

6. Composite Risk Score Analysis

We've computed a **gentrification pressure score** based on: - 40% weight: Income growth (higher = more gentrification pressure) - 30% weight: Population growth (in-migration of higher-earning residents) - 30% weight: Poverty rate decline (displacement of lower-income residents)

[13]:
```
# ============================================================================
# Risk Score Distribution Analysis
# ============================================================================

# Categorize states by risk level
def categorize_risk(score):
    if score >= 75:
        return 'High Risk'
    elif score >= 50:
        return 'Moderate Risk'
    elif score >= 25:
        return 'Low Risk'
    else:
        return 'Minimal Risk'

change_df['risk_category'] = change_df['gentrification_score'].
 ↪apply(categorize_risk)

# Summary by risk category
risk_summary = change_df.groupby('risk_category').agg({
    'state': 'count',
    'income_growth_pct': 'mean',
    'pop_growth_pct': 'mean',
    'poverty_change_pts': 'mean',
}).round(2)
risk_summary.columns = ['Count', 'Avg Income Growth %', 'Avg Pop Growth %',␣
 ↪'Avg Poverty Change pts']

print("  States by Gentrification Risk Category:")
risk_summary
```

```
States by Gentrification Risk Category:
```

[13]:
```
               Count  Avg Income Growth %  Avg Pop Growth %  \
risk_category
High Risk          9               109.26            299.79
Low Risk          22                 9.38            212.64
Minimal Risk       4               -23.61            -66.53
Moderate Risk     17                46.14            213.55

               Avg Poverty Change pts
risk_category
High Risk                       -9.05
Low Risk                         0.38
Minimal Risk                    10.07
Moderate Risk                   -3.22
```

[14]:
```python
# Visualization: Risk Category Pie Chart
category_counts = change_df['risk_category'].value_counts()

fig = px.pie(
    values=category_counts.values,
    names=category_counts.index,
    title='Distribution of States by Gentrification Risk Level',
    color_discrete_sequence=['#CC3311', '#EE7733', '#009988', '#0077BB'],
    template='plotly_white',
)

fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_layout(height=450)
fig.show()
```

7. Spatial Visualization: National Map

Let's create a choropleth map showing gentrification pressure across states.

[15]:
```python
# ============================================================================
# Choropleth Map: Gentrification Pressure by State
# ============================================================================

# Map state names to abbreviations for Plotly
state_abbrev = {
    'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR',␣
↪'California': 'CA',
    'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE', 'Florida': 'FL',␣
↪'Georgia': 'GA',
    'Hawaii': 'HI', 'Idaho': 'ID', 'Illinois': 'IL', 'Indiana': 'IN', 'Iowa':␣
↪'IA',
```

```
    'Kansas': 'KS', 'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME',␣
 ↪'Maryland': 'MD',
    'Massachusetts': 'MA', 'Michigan': 'MI', 'Minnesota': 'MN', 'Mississippi':␣
 ↪'MS', 'Missouri': 'MO',
    'Montana': 'MT', 'Nebraska': 'NE', 'Nevada': 'NV', 'New Hampshire': 'NH',␣
 ↪'New Jersey': 'NJ',
    'New Mexico': 'NM', 'New York': 'NY', 'North Carolina': 'NC', 'North␣
 ↪Dakota': 'ND', 'Ohio': 'OH',
    'Oklahoma': 'OK', 'Oregon': 'OR', 'Pennsylvania': 'PA', 'Rhode Island':␣
 ↪'RI', 'South Carolina': 'SC',
    'South Dakota': 'SD', 'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT',␣
 ↪'Vermont': 'VT',
    'Virginia': 'VA', 'Washington': 'WA', 'West Virginia': 'WV', 'Wisconsin':␣
 ↪'WI', 'Wyoming': 'WY',
    'District of Columbia': 'DC', 'Puerto Rico': 'PR'
}

change_df['state_abbrev'] = change_df['state'].map(state_abbrev)

fig = px.choropleth(
    change_df,
    locations='state_abbrev',
    locationmode='USA-states',
    color='gentrification_score',
    scope='usa',
    color_continuous_scale='RdYlGn_r',
    hover_name='state',
    hover_data={'income_growth_pct': ':.1f', 'pop_growth_pct': ':.1f',␣
 ↪'gentrification_score': ':.0f'},
    title='Gentrification Pressure Index by State (2017-2022)',
)

fig.update_layout(height=500)
fig.show()
```

[16]:
```
# ============================================================================
# Map: Income Growth (Key Gentrification Signal)
# ============================================================================

fig = px.choropleth(
    change_df,
    locations='state_abbrev',
    locationmode='USA-states',
    color='income_growth_pct',
    scope='usa',
    color_continuous_scale='Blues',
```

```
    hover_name='state',
    hover_data={'income_growth_pct': ':.1f', 'income_2022': ':,.0f'},
    title='Median Household Income Growth by State (2017-2022)',
)

fig.update_layout(height=500)
fig.show()
```

8. Temporal Analysis: Economic Leading Indicators

Let's visualize the economic context over time to understand the macro pressures.

[17]:
```
# ============================================================================
# Time Series: Economic Context
# ============================================================================

fig = make_subplots(
    rows=2, cols=1,
    subplot_titles=('Housing Starts (New Construction)', '30-Year Mortgage␣
↪Rate'),
    vertical_spacing=0.12,
)

# Housing starts
fig.add_trace(
    go.Scatter(
        x=housing_starts.index,
        y=housing_starts['value'],
        fill='tozeroy',
        fillcolor='rgba(0, 119, 187, 0.2)',
        line=dict(color='#0077BB', width=2),
        name='Housing Starts'
    ),
    row=1, col=1
)

# Mortgage rates
fig.add_trace(
    go.Scatter(
        x=mortgage_rates.index,
        y=mortgage_rates['value'],
        line=dict(color='#CC3311', width=2),
        name='Mortgage Rate'
    ),
    row=2, col=1
)

fig.update_layout(
```

```python
    title='Economic Context: Housing Market Conditions (2017-2022)',
    template='plotly_white',
    height=550,
    showlegend=False,
)

fig.update_yaxes(title_text="Units (000s)", row=1, col=1)
fig.update_yaxes(title_text="Rate (%)", row=2, col=1)

fig.show()
```

[18]:
```python
# ============================================================================
# Correlation Analysis: Economic Factors and Demographic Change
# ============================================================================

# Note: This is a simplified demonstration. Full analysis would require
# state-level economic data (Professional tier) for proper correlation.

print(" Key Observations from Economic Context:")
print("=" * 55)
print()
print("1. HOUSING SUPPLY:")
housing_2017 = housing_starts.loc['2017'].mean()['value']
housing_2022 = housing_starts.loc['2022'].mean()['value']
print(f"   • 2017 avg: {housing_2017:.0f}k units/month")
print(f"   • 2022 avg: {housing_2022:.0f}k units/month")
print(f"   • Change: {((housing_2022/housing_2017)-1)*100:+.1f}%")

print()
print("2. FINANCING COSTS:")
mortgage_2017 = mortgage_rates.loc['2017'].mean()['value']
mortgage_2022 = mortgage_rates.loc['2022'].mean()['value']
print(f"   • 2017 avg rate: {mortgage_2017:.2f}%")
print(f"   • 2022 avg rate: {mortgage_2022:.2f}%")
print(f"   • Monthly payment impact: ~${((mortgage_2022 - mortgage_2017) * 3):,.
 ↪0f} on $300k home")

print()
print("3. IMPLICATION:")
print("   Rising rates + limited supply = affordability pressure")
print("   This creates displacement risk in gentrifying areas")
```

```
 Key Observations from Economic Context:
=======================================================

1. HOUSING SUPPLY:
   • 2017 avg: 1205k units/month
   • 2022 avg: 1552k units/month
```

```
    • Change: +28.8%

2. FINANCING COSTS:
   • 2017 avg rate: 3.99%
   • 2022 avg rate: 5.34%
   • Monthly payment impact: ~$4 on $300k home

3. IMPLICATION:
   Rising rates + limited supply = affordability pressure
   This creates displacement risk in gentrifying areas
```

9. Key Insights and Policy Implications

[19]:

```python
# ============================================================================
# Key Insights Summary
# ============================================================================

# Get top 5 states
top_5 = change_df.head(5)

print("=" * 65)
print("  KEY INSIGHTS: Gentrification Early Warning Analysis")
print("=" * 65)
print(f"\n  Analysis Period: 2017-2022 (Census ACS 5-Year Estimates)")
print(f"  Geographic Scope: 50 States + DC (Community Tier = State-level)")

print(f"\n  TOP 5 STATES BY GENTRIFICATION PRESSURE:")
for _, row in top_5.iterrows():
    print(f"   {row['rank']}. {row['state']}")
    print(f"      • Income Growth: +{row['income_growth_pct']:.1f}%")
    print(f"      • Population Growth: +{row['pop_growth_pct']:.1f}%")
    print(f"      • Score: {row['gentrification_score']:.0f}/100")

print(f"\n  NATIONAL AVERAGES:")
print(f"   • Mean Income Growth: {change_df['income_growth_pct'].mean():.1f}%")
print(f"   • Mean Population Growth: {change_df['pop_growth_pct'].mean():.1f}%")
print(f"   • Mean Poverty Change: {change_df['poverty_change_pts'].mean():+.2f}␣
↪pts")

print("\n" + "=" * 65)
print("  POLICY IMPLICATIONS")
print("=" * 65)
print("""
1. DISPLACEMENT RISK: States with high income growth but
   limited housing construction face greatest displacement risk.

2. EARLY WARNING SIGNALS:
   - Rapid income increases (>30% in 5 years)
```

```
   - Population growth exceeding housing starts
   - Declining poverty (may indicate displacement)

3. INTERVENTION POINTS:
   - Tenant protections before rapid change
   - Affordable housing preservation
   - Community land trusts in high-risk areas

4. LIMITATIONS:
   - State-level masks neighborhood variation
   - Upgrade to Professional tier for tract-level analysis
   - Add rental/business data for leading indicators
""")
```

```
==================================================================
 KEY INSIGHTS: Gentrification Early Warning Analysis
==================================================================
```

Analysis Period: 2017-2022 (Census ACS 5-Year Estimates)
Geographic Scope: 50 States + DC (Community Tier = State-level)

TOP 5 STATES BY GENTRIFICATION PRESSURE:

1. New York
   - Income Growth: +311.6%
   - Population Growth: +476.4%
   - Score: 94/100
2. New Hampshire
   - Income Growth: +106.2%
   - Population Growth: +-24.9%
   - Score: 82/100
3. Massachusetts
   - Income Growth: +97.8%
   - Population Growth: +42.7%
   - Score: 81/100
4. Utah
   - Income Growth: +86.6%
   - Population Growth: +-25.8%
   - Score: 79/100
5. California
   - Income Growth: +65.8%
   - Population Growth: +1262.9%
   - Score: 78/100

NATIONAL AVERAGES:

- Mean Income Growth: 36.1%
- Mean Population Growth: 206.5%
- Mean Poverty Change: -1.69 pts

```
================================================================
  POLICY IMPLICATIONS
================================================================

1. DISPLACEMENT RISK: States with high income growth but
   limited housing construction face greatest displacement risk.

2. EARLY WARNING SIGNALS:
   - Rapid income increases (>30% in 5 years)
   - Population growth exceeding housing starts
   - Declining poverty (may indicate displacement)

3. INTERVENTION POINTS:
   - Tenant protections before rapid change
   - Affordable housing preservation
   - Community land trusts in high-risk areas

4. LIMITATIONS:
   - State-level masks neighborhood variation
   - Upgrade to Professional tier for tract-level analysis
   - Add rental/business data for leading indicators
```

10. Next Steps

1.2.1 Upgrade to Professional Tier for Tract-Level Analysis

For neighborhood-level gentrification early warning, upgrade to **Professional Tier** ($149-599/mo):

```python
from krl_data_connectors.professional import (
    CensusACSConnector,      # Tract/block group demographics
    ZillowConnector,         # Neighborhood rent/value changes
)
from krl_geospatial import QueenWeights, clustering

# Tract-level analysis
census_pro = CensusACSConnector(license_key="YOUR_KEY")
tracts = census_pro.get_demographics(
    geography="tract",
    state="CA",
    county="075"  # San Francisco
)

# Spatial clustering to identify hotspots
weights = QueenWeights(tracts)
clusters = clustering.lisa(tracts['income_growth'], weights)
```

1.2.2 Related Notebooks

- **01-metro-housing-wage-divergence.ipynb**: Economic pressure analysis
- **04-environmental-justice-health.ipynb**: Pollution burden mapping
- **10-urban-resilience-dashboard.ipynb**: Complete multi-source workflow

11. Data Provenance

[20]:
```
# ============================================================================
# Data Provenance Documentation
# ============================================================================

provenance = """
## Data Sources

| Dataset | Source | Description |
|---------|--------|-------------|
| Demographics | Census ACS 5-Year | Population, income, poverty, race/
↪ethnicity |
| Housing Starts | FRED (HOUST) | New residential construction |
| Mortgage Rates | FRED (MORTGAGE30US) | 30-year fixed rate mortgage |
| CPI | FRED (CPIAUCSL) | Consumer Price Index |

## Census ACS Variables Used

| Code | Description |
|------|-------------|
| B01001_001E | Total Population |
| B19013_001E | Median Household Income |
| B17001_002E | Population Below Poverty Level |
| B02001_002E | White Alone |
| B02001_003E | Black or African American Alone |
| B02001_005E | Asian Alone |
| B03003_003E | Hispanic or Latino |

## Access Method

- **Connector Package**: `krl_data_connectors` v1.0.0
- **Tier**: Community (Free)
- **API Keys Required**: None
- **Geographic Level**: State (Community tier limit)

## Reproducibility

```python
from krl_data_connectors.community import CensusACSPublicConnector,␣
↪FREDBasicConnector
```
```

```
census = CensusACSPublicConnector()
fred = FREDBasicConnector()

demographics = census.get_demographics_by_state(year=2022)
housing = fred.get_series("HOUST", start_date="2017-01-01")
```
"""

from IPython.display import Markdown
Markdown(provenance)
```

[20]:

### 1.3 Data Sources

| Dataset | Source | Description |
|---|---|---|
| Demographics | Census ACS 5-Year | Population, income, poverty, race/ethnicity |
| Housing Starts | FRED (HOUST) | New residential construction |
| Mortgage Rates | FRED (MORTGAGE30US) | 30-year fixed rate mortgage |
| CPI | FRED (CPIAUCSL) | Consumer Price Index |

### 1.4 Census ACS Variables Used

| Code | Description |
|---|---|
| B01001_001E | Total Population |
| B19013_001E | Median Household Income |
| B17001_002E | Population Below Poverty Level |
| B02001_002E | White Alone |
| B02001_003E | Black or African American Alone |
| B02001_005E | Asian Alone |
| B03003_003E | Hispanic or Latino |

### 1.5 Access Method

- **Connector Package**: `krl_data_connectors` v1.0.0
- **Tier**: Community (Free)
- **API Keys Required**: None
- **Geographic Level**: State (Community tier limit)

### 1.6 Reproducibility

```
from krl_data_connectors.community import CensusACSPublicConnector, FREDBasicConnector

census = CensusACSPublicConnector()
fred = FREDBasicConnector()

demographics = census.get_demographics_by_state(year=2022)
housing = fred.get_series("HOUST", start_date="2017-01-01")
```
```

[21]:
```python
# ============================================================================
# Session Information for Reproducibility
# ============================================================================

import sys

print("  Session Information")
print("=" * 50)
print(f"Python Version: {sys.version}")
print(f"Pandas Version: {pd.__version__}")
print(f"NumPy Version: {np.__version__}")
print()
print("  KRL Suite Packages Used:")
print("   • krl_data_connectors.community.CensusACSPublicConnector")
print("   • krl_data_connectors.community.FREDBasicConnector")
print("   • krl_core (Logging)")
print()
print(f"  Execution Completed: {datetime.now().isoformat()}")
```

```
  Session Information
==================================================
Python Version: 3.13.7 (main, Aug 14 2025, 11:12:11) [Clang 17.0.0
(clang-1700.0.13.3)]
Pandas Version: 2.3.3
NumPy Version: 2.3.4

  KRL Suite Packages Used:
   • krl_data_connectors.community.CensusACSPublicConnector
   • krl_data_connectors.community.FREDBasicConnector
   • krl_core (Logging)

  Execution Completed: 2025-11-27T12:01:33.321445
```

1.7 About the KRL Suite

Package	Description	This Notebook
`krl-data-connectors`	67+ economic data connectors	Census ACS, FRED
`krl-model-zoo`	Regional & forecasting models	(Not used)
`krl-geospatial-tools`	Spatial analysis & mapping	(Pro tier for tracts)
`krl-causal-policy-toolkit`	Causal inference methods	(Pro tier)
`krl-open-core`	Shared utilities & logging	Logging

Learn More: github.com/KR-Labs
