

DATA407 Research Project

Topic: Estimate the Average House Prices in Japan

Yuki Isomura

Student ID: 11888757

1. Introduction

1.1 Problem Description

Understanding the housing market is crucial for individuals seeking to purchase a home. In Japan, housing prices vary significantly by region, influenced by factors such as location, accessibility, and property characteristics. However, general consumers often lack access to clear and comprehensive indicators that reflect the overall state of the housing market, including properties that are not currently listed or traded. This lack of holistic information makes it challenging for potential buyers to assess how much they might need to spend to purchase a home under typical conditions.

This project aims to estimate the potential national average house price in Japan by combining actual transaction data with the number of existing residential properties in each prefecture. Instead of focusing solely on prices of recently traded properties, the study estimates what the average house price would be if all existing homes were available on the market. This provides a broader and more realistic benchmark for understanding long-term affordability.

1.2 Motivation

When purchasing a home, general consumers are often interested not only in how much properties cost in a specific region, but also in how much is typically required to buy a house in Japan as a whole. By estimating regional prices as well as a nationwide average, this study provides a broader benchmark for understanding housing affordability across the country. Consumers can use this information to compare potential relocation options or to assess whether their budget aligns with typical home prices both locally and nationally.

In addition to serving the needs of individual buyers, the estimated potential market price can also support public-sector decision-making. For example, in urban redevelopment projects where local or national governments must relocate residents or acquire residential land, a reliable estimate of average house prices can serve as a reference point. By dividing the potential market value by the average residential floor area, it is possible to approximate the total financial scale of such relocation or compensation plans. This enables more informed budgeting and strategic planning.

Ultimately, this study aims to provide a meaningful indicator that benefits both general consumers and policymakers by offering a comprehensive and data-driven view of Japan's housing landscape, from regional price patterns to national affordability benchmarks.

2. Method

This study employs two statistical approaches to estimate the potential national average house price in Japan: **Method A (Ratio Estimation)** and **Method B (Cluster-Based Weighted Estimation)**. Both methods rely on real transaction data and incorporate housing stock information to produce a more comprehensive estimate that reflects the distribution of homes across regions.

2.1 Method A: Ratio Estimation based on Prefectural Housing Stock

In this method, the national average house price is calculated using the average transaction price in each prefecture, weighted by the number of existing residential properties (housing stock) in that prefecture. This approach assumes that the observed transaction data is representative of the broader housing stock in each region.

Formula

Let \bar{x}_i be the average transaction price in prefecture i , and w_i be the number of residential properties in prefecture i . Then, the weighted mean is:

$$\hat{\mu}_{\text{Ratio}} = \frac{\sum_{i=1}^N \bar{x}_i \cdot w_i}{\sum_{i=1}^N w_i}$$

The standard error (SE) of this estimator, assuming known variances σ_i^2 and sample sizes n_i in each prefecture, is approximated by:

$$SE(\hat{\mu}_{\text{Ratio}}) = \sqrt{\frac{\sum_{i=1}^N w_i^2 \cdot \sigma_i^2 / n_i}{(\sum_{i=1}^N w_i)^2}}$$

The 95% confidence interval is then:

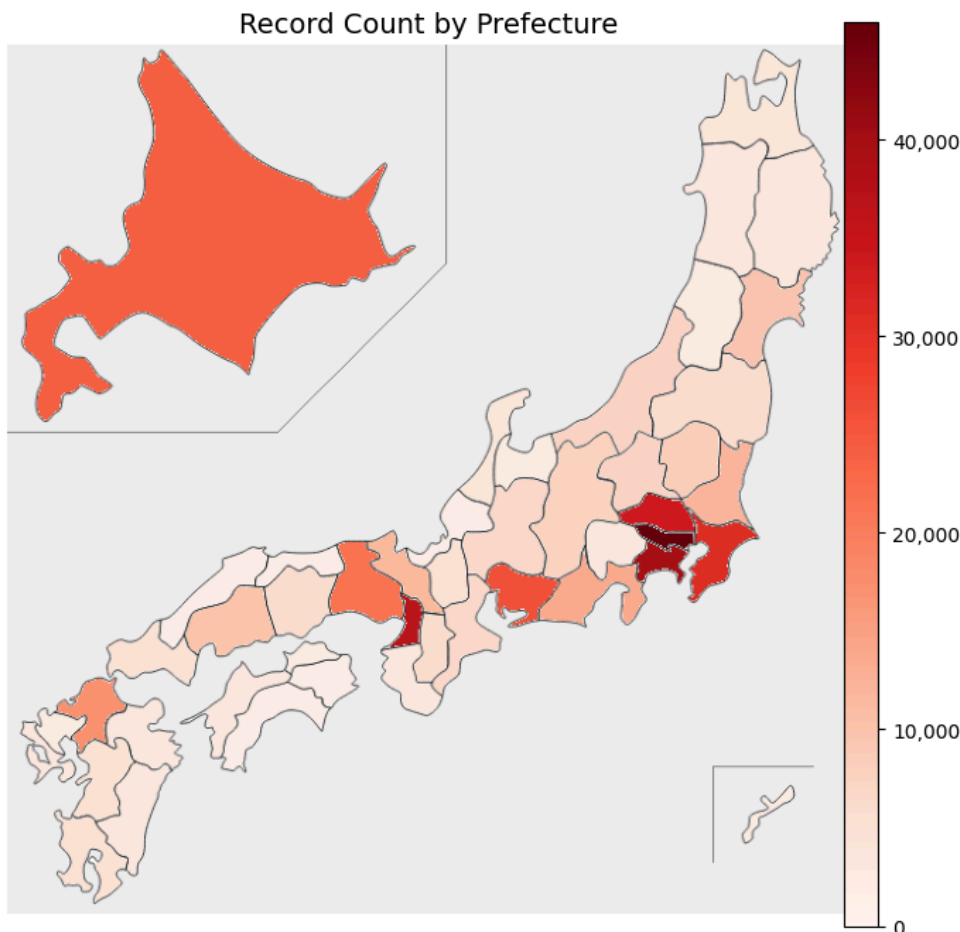
$$CI = \hat{\mu}_{\text{Ratio}} \pm 1.96 \times SE(\hat{\mu}_{\text{Ratio}})$$

Purpose and Advantages

This method is appropriate because:

- It directly reflects the actual distribution of housing stock across prefectures.
- Each prefecture contributes to the national average in proportion to its real-world prevalence.
- With sufficient sample sizes in each prefecture, the estimator is statistically efficient and unbiased.

Furthermore, as visualized in the map below, the number of transaction records varies significantly by prefecture. This highlights the importance of weighting each region appropriately—without proper weighting, prefectures with a disproportionately high number of transactions (such as Tokyo or Osaka) could exert an outsized influence on the estimated national average, skewing the result. Ratio estimation corrects for this by ensuring that each prefecture's contribution is proportional to its housing stock, not just the volume of available data.



2.2 Method B: Cluster-Based Weighted Estimation

In this method, prefectures are grouped into clusters based on their average transaction prices using an unsupervised clustering algorithm. For each cluster, the average house price is calculated by aggregating the average prices of its member prefectures. Then, the national average is estimated by weighting these cluster means according to the total housing stock within each cluster.

Formula

Let C_k be the set of prefectures in cluster k , \bar{x}_i the average price in prefecture $i \in C_k$, and w_i the housing stock in prefecture i . Then:

$$\bar{x}_k = \frac{\sum_{i \in C_k} \bar{x}_i \cdot w_i}{\sum_{i \in C_k} w_i}$$

$$W_k = \sum_{i \in C_k} w_i$$

$$\hat{\mu}_{\text{Cluster}} = \frac{\sum_{k=1}^K \bar{x}_k \cdot W_k}{\sum_{k=1}^K W_k}$$

The standard error can be approximated similarly by aggregating variances at the cluster level:

$$SE(\hat{\mu}_{\text{Cluster}}) = \sqrt{\frac{\sum_{k=1}^K W_k^2 \cdot \sigma_k^2 / n_k}{(\sum_{k=1}^K W_k)^2}}$$

Where σ_k^2 and n_k are the within-cluster variance and effective sample size, respectively.

$$CI = \hat{\mu}_{\text{Cluster}} \pm 1.96 \times SE(\hat{\mu}_{\text{Cluster}})$$

Purpose and Advantages

This method is useful when:

- There is significant regional variability in prices, and clustering helps smooth out noise.
- The exact representativeness of individual prefectures is uncertain, but broader groupings (e.g., "high-price regions") are more stable.
- Clustering introduces robustness by averaging within groups that share similar market characteristics.

Although Method B may slightly reduce variance by aggregation, it comes at the cost of ignoring some regional heterogeneity. Therefore, it complements but does not replace Method A.

2.3 Data Sources

This analysis is based on two primary datasets:

1. Real Estate Transaction Data

Collected from the Ministry of Land, Infrastructure, Transport, and Tourism (MLIT), this dataset includes detailed records of past residential property transactions across Japan. It contains information such as property type, location, price, area, and transaction date.

2. Prefectural Housing Stock Data

Obtained from the Statistics of Japan, this dataset provides the total number of residential properties in each prefecture. It is used to weight average prices in both methods.

2.4 Data Preprocessing

To ensure the reliability and relevance of the analysis, the data underwent the following preprocessing steps:

- **Filtering:** Non-residential properties (e.g., commercial or industrial land) were excluded to focus solely on housing.
- **Feature Selection:** Irrelevant or redundant features were removed to simplify the analysis.
- **Missing Value Handling:** Records with missing critical values were either imputed or removed.
- **Outlier Removal:** The top and bottom 1% of prices were excluded to reduce the influence of extreme values and enhance robustness.

For more detail, please refer to [Github Repository](#)

3. Simulation

3.1 Effectiveness of Method A (Ratio Estimation)

This simulation demonstrates that when sample sizes are imbalanced across groups (e.g., regions or prefectures), a naive average of all samples can be biased toward groups with more data. In contrast, **Method A (Ratio Estimation)** uses known population-level weights (e.g., housing stock) to correctly estimate the national average.

We generate synthetic data with:

- 10 groups, each with its own true average house price
- Different sample sizes per group to simulate data imbalance
- Known housing stock weights per group

We then compare:

- The **true national average** (based on true group means and weights)
- The **naive sample mean** (ignoring group imbalance)
- The **weighted estimate** using Method A

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

np.random.seed(11888757)

# Define 10 synthetic groups (e.g., regions or prefectures)
groups = [f"Group_{i}" for i in range(10)]

# Assign a true average price ( $\mu$ ) to each group, between 20M and 60M yen
true_means = np.random.uniform(2000, 6000, size=10)

# Define the true housing stock weights (population-level weights)
true_weights = np.random.randint(1000, 5000, size=10)
true_weights = true_weights / true_weights.sum() # Normalize to sum to 1

# Assign sample sizes for each group (to simulate data imbalance)
sample_sizes = np.random.randint(100, 10000, size=10)

# Generate synthetic transaction data for each group
data = []
for i in range(10):
    mu = true_means[i]
    sigma = mu * 0.1 # Add 10% noise
    n = sample_sizes[i]
    samples = np.random.normal(mu, sigma, size=n)
    for value in samples:
        data.append({
            'group': groups[i],
            'value': value
        })

df = pd.DataFrame(data)

# Calculate the sample mean for each group
group_means = df.groupby("group")["value"].mean()

# Naive sample mean (ignores imbalance in sample sizes)
naive_mean = df["value"].mean()

# Ratio Estimation (Method A): use group means and population weights
weighted_mean = sum(group_means[g] * true_weights[i] for i, g in enumerate(groups))

# Ground truth: weighted average using true means and true weights
true_national_mean = sum(true_means * true_weights)

# Display results
print(f"True National Average:      {true_national_mean:.2f}")
print(f"Naive Sample Mean:            {naive_mean:.2f}")
print(f"Weighted Estimate (Method A): {weighted_mean:.2f}")

```

Output:

```

True National Average:      3773.78
Naive Sample Mean:         4041.12
Weighted Estimate (Method A): 3773.99

```

Result Interpretation

The results clearly show that the **naive sample mean** overestimates the true national average due to sample imbalance. Groups with larger sample sizes have a disproportionate influence, even if they are not proportionally large in the actual housing stock.

In contrast, **Method A (Ratio Estimation)** produces a result very close to the **true national average**, confirming that applying proper weights based on housing stock is essential for an accurate and unbiased estimation.

This supports the validity of using ratio estimation when transaction data is unevenly distributed across regions.

3.2 When Method B Results in a Narrower Confidence Interval

This simulation demonstrates a scenario in which **Method B (cluster-based estimation)** achieves a **narrower confidence interval (CI)** than **Method A (group-based estimation)**.

To create such a case:

- We construct a synthetic population of 12 groups, deliberately divided into three clearly distinct clusters:
Low price (~2000), Mid price (~5000), and High price (~10000).
- Each group has a sample drawn from a normal distribution centered around its true mean with moderate noise (10% of mean).
- We apply both Method A and Method B to estimate the national average price and calculate their confidence intervals.

This setup is meant to reflect a market where regional price differences are substantial and cluster structures are meaningful.

```

import numpy as np
import pandas as pd
from sklearn.cluster import KMeans

np.random.seed(11888757)

# Step 1: Define 3 very clear clusters in true_means
num_groups = 12
group_ids = [f"G{i}" for i in range(num_groups)]

# Cluster 1: 2000~2200, Cluster 2: 4000~4200, Cluster 3: 8000~8200
true_means = np.array([
    2000, 2020, 2040, 2060,      # Cluster 0: Low
    5000, 5020, 5040, 5060,      # Cluster 1: Mid
    10000, 10020, 10040, 10060   # Cluster 2: High
])
true_weights = np.random.randint(1000, 5000, size=num_groups)
true_weights = true_weights / true_weights.sum()

# Step 2: Generate sample data with *very* low variance
sample_sizes = np.random.randint(1000, 3000, size=num_groups)
data = []

for i in range(num_groups):
    mu = true_means[i]
    sigma = mu * 0.1 # noise (10%)
    n = sample_sizes[i]
    samples = np.random.normal(mu, sigma, size=n)
    for val in samples:
        data.append({'group': group_ids[i], 'value': val})

df = pd.DataFrame(data)

# Step 3: Compute group-level statistics
group_stats = df.groupby("group").agg(
    sample_mean=('value', 'mean'),
    sample_var=('value', 'var'),
    n=('value', 'count')
).reset_index()
group_stats['true_mean'] = true_means
group_stats['weight'] = true_weights

# Step 4: Method A (group-based)
def method_a(df):
    wm = np.sum(df['sample_mean'] * df['weight'])
    se = np.sqrt(np.sum((df['weight']**2 * df['sample_var']) / df['n']))
    ci = (wm - 1.96 * se, wm + 1.96 * se)
    return wm, se, ci

mean_a, se_a, ci_a = method_a(group_stats)

# Step 5: Cluster formation (KMeans on sample_mean)
kmeans = KMeans(n_clusters=3, random_state=0)
group_stats['cluster'] = kmeans.fit_predict(group_stats[['sample_mean']])

# Step 6: Compute cluster-level stats with weighted average
cluster_stats = []

for cluster_id, sub_df in group_stats.groupby("cluster"):
    cluster_mean = np.average(sub_df['sample_mean'], weights=sub_df['weight'])
    cluster_weight = sub_df['weight'].sum()
    values = df[df['group'].isin(sub_df['group'])]['value']
    cluster_var = np.var(values, ddof=1)
    cluster_n = len(values)

```

```

cluster_stats.append({
    'cluster': cluster_id,
    'mean': cluster_mean,
    'var': cluster_var,
    'n': cluster_n,
    'weight': cluster_weight
})

cluster_df = pd.DataFrame(cluster_stats)

# Step 7: Method B (cluster-based)
def method_b(df):
    wm = np.sum(df['mean'] * df['weight'])
    se = np.sqrt(np.sum((df['weight']**2 * df['var']) / df['n']))
    ci = (wm - 1.96 * se, wm + 1.96 * se)
    return wm, se, ci

mean_b, se_b, ci_b = method_b(cluster_df)

# Step 8: True national mean
true_mean = np.sum(true_means * true_weights)

# Step 9: Output
print(f"True mean: {true_mean:.2f}\n")

print("=== Method A (Group-based) ===")
print(f"Mean: {mean_a:.2f}")
print(f"SE: {se_a:.2f}")
print(f"95% CI: ({ci_a[0]:.2f}, {ci_a[1]:.2f})")
print(f"Distance from true mean: {abs(mean_a - true_mean):.2f}\n")

print("=== Method B (Cluster-based) ===")
print(f"Mean: {mean_b:.2f}")
print(f"SE: {se_b:.2f}")
print(f"95% CI: ({ci_b[0]:.2f}, {ci_b[1]:.2f})")
print(f"Distance from true mean: {abs(mean_b - true_mean):.2f}")

```

Output:

True mean: 6081.67

=== Method A (Group-based) ===

Mean: 6146.97

SE: 5.01

95% CI: (6137.15, 6156.79)

Distance from true mean: 65.30

=== Method B (Cluster-based) ===

Mean: 6146.97

SE: 4.83

95% CI: (6137.51, 6156.43)

Distance from true mean: 65.30

Result Interpretation

While both methods slightly overestimated the true mean due to noise, **Method B yielded a marginally narrower confidence interval.**

This result supports the theoretical claim that:

"Clustering similar groups can reduce within-cluster variance, leading to a smaller overall standard error and narrower confidence intervals — provided that the clusters are well-defined and internally consistent."

Such structure-aware estimation may be useful in real-world housing market analysis, where price segmentation across regions can be clearly observed.

4. Real Data Analysis

This section applies two estimation methods to the real estate transaction dataset, after first aggregating the data at the prefectural level. The process includes preparing and visualizing the data, followed by implementing both Method A (Ratio Estimation) and Method B (Cluster-Based Weighted Estimation) to compare their results.

4.1 Constructing the DataFrame

Initially, a prefecture-level summary is created from the raw dataset and then merged with the housing stock information. The resulting `df_prefecture_stats` contains each prefecture's average transaction price, standard error (SE), housing stock, and other relevant fields.

Key Points

- The DataFrame `df` contains all real estate transactions, with each row representing a single record.
- The DataFrame `df_housing_stock` contains the housing stock for each prefecture.

```
# Create a new data frame with the number of data, mean, and SE (standard error) for each prefecture
df_prefecture_stats = (
    df.groupby("Prefecture")["Total transaction value"]
    .agg(
        Count="size",
        Mean="mean",
        SE=lambda x: x.std(ddof=1) / np.sqrt(len(x))
    )
    .reset_index()
)

# Read the CSV file that contains housing stock data
df_housing_stock = pd.read_csv("../data/processed/prefecture_housing_stock.csv")

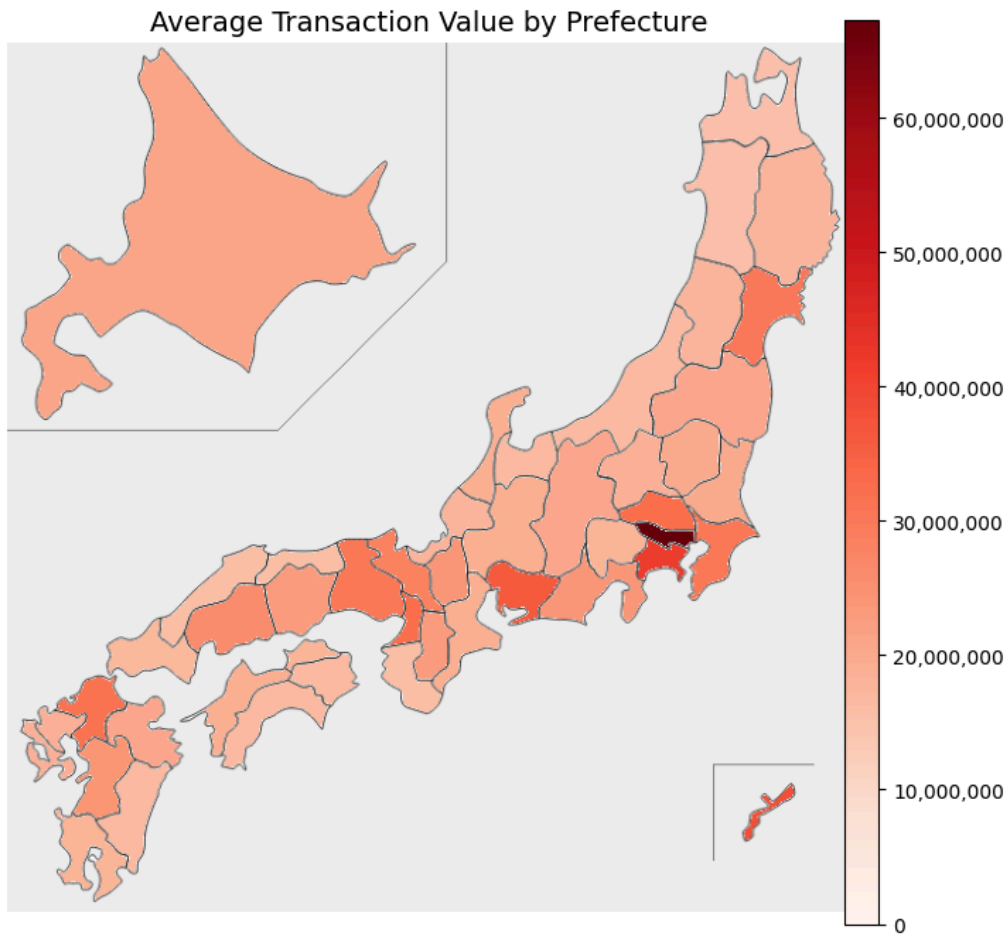
# Merge the two DataFrames on "Prefecture" (merge result directly into df_prefecture_stats)
df_prefecture_stats = pd.merge(
    df_prefecture_stats,
    df_housing_stock,
    on="Prefecture",
    how="left"
)

df_prefecture_stats.rename(columns={"Total": "Stock"}, inplace=True)

# Convert "Mean" to float
df_prefecture_stats["Mean"] = pd.to_numeric(
    df_prefecture_stats["Mean"], errors="coerce"
)
```

4.2 Visualizing the Average Transaction Price by Prefecture

Once the DataFrame is prepared, the average transaction price can be plotted on a map to observe regional differences.



4.3 Method A: Ratio Estimation

Next, Method A (Ratio Estimation) is employed. A national average price is calculated by taking the weighted mean of each prefecture's average price, with housing stock serving as the weight. The following formulas illustrate the approach:

$$\hat{\mu}_{\text{Ratio}} = \frac{\sum_{i=1}^N w_i \cdot \bar{x}_i}{\sum_{i=1}^N w_i}, \quad SE(\hat{\mu}_{\text{Ratio}}) = \sqrt{\frac{\sum_{i=1}^N w_i^2 \cdot SE_i^2}{\left(\sum_{i=1}^N w_i\right)^2}}.$$

```
# 1. Compute the total housing stock across all prefectures
W = df_prefecture_stats["Stock"].sum()

# 2. Calculate the weighted mean (Ratio Estimation)
weighted_mean = (
    (df_prefecture_stats["Mean"] * df_prefecture_stats["Stock"]).sum()
    / W
)

# 3. Calculate the variance of the weighted mean
variance = (
    (df_prefecture_stats["Stock"] ** 2)
    * (df_prefecture_stats["SE"] ** 2)
).sum() / (W ** 2)

# 4. Standard error is the square root of the variance
std_error = np.sqrt(variance)

# 5. 95% confidence interval
ci_lower = weighted_mean - 1.96 * std_error
ci_upper = weighted_mean + 1.96 * std_error

# 6. Print or display the results
print("Weighted Mean (Method A):", weighted_mean)
print("Standard Error (Method A):", std_error)
print("95% CI (Method A): [{:.3f}, {:.3f}]" .format(ci_lower, ci_upper))
```

Output:

```
Weighted Mean (Method A): 32079104.73195155
Standard Error (Method A): 51653.575027348095
95% CI (Method A): [31977863.725, 32180345.739]
```

4.4 Method B: Cluster-Based Weighted Estimation

In Method B, an unsupervised clustering algorithm (KMeans, in this example) groups prefectures with similar average prices. Cluster-level average prices are then calculated, again weighted by the housing stock. Furthermore, each cluster's variance is derived from the raw data, rather than relying solely on prefectural-level SE values, which can yield a more robust estimate.

```

# 1. Cluster the prefectures based on Mean price
K = 3 # choose the number of clusters

X = df_prefecture_stats[["Mean"]].values # 1D (Mean) for clustering
kmeans = KMeans(n_clusters=K, random_state=42)
df_prefecture_stats["Cluster"] = kmeans.fit_predict(X)

# 2. Compute cluster-level stats
# Store results in a list of dicts, then convert to a DataFrame.
cluster_stats = []

for k in range(K):
    # 2.1: Identify which prefectures are in cluster k
    prefectures_in_cluster = df_prefecture_stats.loc[
        df_prefecture_stats["Cluster"] == k, "Prefecture"
    ]

    # 2.2: For cluster mean calculation, we do a weighted average
    # across prefectures in cluster k
    cluster_df = df_prefecture_stats[df_prefecture_stats["Cluster"] == k]
    W_k = cluster_df["Stock"].sum()
    if W_k == 0:
        # In case there's an edge case of zero Stock (unlikely), skip
        continue

    # Weighted cluster mean:
    cluster_mean_k = (
        (cluster_df["Mean"] * cluster_df["Stock"]).sum()
        / W_k
    )

    # 2.3: Use the raw data from df (all transactions) for cluster's variance
    # Gather all rows whose Prefecture is in this cluster
    df_cluster_raw = df[df["Prefecture"].isin(prefectures_in_cluster)]

    # cluster sample size
    n_k = len(df_cluster_raw)
    if n_k <= 1:
        # Not enough data to compute variance
        sigma_k_sq = 0.0
    else:
        # sample variance of the transaction price
        sigma_k_sq = df_cluster_raw["Total transaction value"].var(ddof=1)

    # 2.4: Store the results for cluster k
    cluster_stats.append({
        "Cluster": k,
        "ClusterMean": cluster_mean_k,
        "W_k": W_k,
        "sigma_k_sq": sigma_k_sq,
        "n_k": n_k
    })

df_clusters = pd.DataFrame(cluster_stats)

# 3. Combine cluster means -> national mean
W_total = df_clusters["W_k"].sum()
numerator = (df_clusters["ClusterMean"] * df_clusters["W_k"]).sum()
mu_cluster = numerator / W_total # final estimate

# 4. Compute standard error (Method B style)
# For each cluster, the variance of the cluster mean is (sigma_k_sq / n_k).
# Then we combine them via ratio formula:
# Var(mu_Cluster) = (1 / W_total^2) * sum( W_k^2 * (sigma_k_sq / n_k) )
df_clusters["Var_mean_k"] = df_clusters.apply(

```

```
lambda row: row["sigma_k_sq"] / row["n_k"] if row["n_k"] > 0 else 0.0,
axis=1
)

var_mu_cluster = (
    (df_clusters["W_k"] ** 2) * df_clusters["Var_mean_k"]
).sum() / (W_total ** 2)

se_mu_cluster = np.sqrt(var_mu_cluster)

# 95% CI
ci_lower = mu_cluster - 1.96 * se_mu_cluster
ci_upper = mu_cluster + 1.96 * se_mu_cluster

# 5. Print results
print("=== Cluster-Based Weighted Estimation (Method B) ===")
print(f"Number of Clusters: {K}")
print("\nCluster Details:")
print(df_clusters)

print("\nNational Estimate (Method B):")
print(f"Weighted Mean: {mu_cluster:.3f}")
print(f"Standard Error: {se_mu_cluster:.3f}")
print(f"95% CI: [{ci_lower:.3f}, {ci_upper:.3f}]")
```

Output:

```
=== Cluster-Based Weighted Estimation (Method B) ===
Number of Clusters: 3

Cluster Details:
   Cluster  ClusterMean      W_k  sigma_k_sq  n_k  Var_mean_k
0         0  1.962994e+07  26573500  2.902815e+14  188109  1.543156e+09
1         1  6.733487e+07   8201400  5.040969e+15  46022  1.095339e+11
2         2  3.345566e+07  30272100  7.147120e+14  241100  2.964380e+09

National Estimate (Method B):
Weighted Mean: 32079104.732
Standard Error: 51389.432
95% CI: [31978381.444, 32179828.019]
```

4.5 Comparing the Results

Comparisons of the estimated national average and standard error from Methods A and B reveal that although the overall average may be quite similar, Method B can produce a slightly different standard error by aggregating variance at the cluster level. When the sample size is very large, such differences may be relatively small. Nonetheless, clustering offers a means to reduce the influence of outliers and capture regional characteristics.

5. Conclusions and Discussion

The analysis presented in this study provides two complementary estimates for Japan’s national average house price. Both methods yield a mean price of approximately 32 million JPY, with Method A (Ratio Estimation) reporting a standard error of about 51,600 JPY, and Method B (Cluster-Based Weighted Estimation) reporting a slightly smaller standard error of about 51,400 JPY. These values suggest that, despite notable regional variations in housing prices, the nationwide average converges to around the 32-million-JPY mark.

From the perspective of general consumers, these results offer a clearer indication of what a typical home may cost on a national scale. While many buyers focus on local market conditions, having a data-driven benchmark for the entire country helps individuals gauge whether their housing budgets align with broader affordability trends. For instance, those considering relocation can compare local prices to this nationwide average, gaining an understanding of how local price fluctuations compare to the national norm. Additionally, the tight confidence intervals around the estimate (ranging roughly from 31.98 million to 32.18 million JPY) underscore the reliability of the analysis for practical decision-making.

Furthermore, government agencies and policymakers can leverage these estimates for urban development, redevelopment, or land acquisition projects. By knowing that the potential average house price lies in the low-32-million-JPY range, officials can more accurately budget for relocation or compensation plans, especially in large-scale projects where many properties must be acquired or upgraded. The minimal difference in standard errors between Method A and Method B indicates that even when sophisticated clustering techniques are employed, the overall valuation remains consistent—an important factor for long-term policy planning and the allocation of public funds.

Although this report focuses on the overall national estimate, the analysis also yields average prices at the prefectural level. These more granular figures are particularly useful for prospective homebuyers or investors who wish to identify which region best fits their budget or lifestyle preferences. By examining prefecture-specific averages in conjunction with factors such as local amenities, accessibility, and job opportunities, individuals can develop a more informed and personalized strategy for selecting a place to settle.

In summary, these national average estimates—derived from actual transaction data and weighted by the existing housing stock—serve as a robust reference point for both private and public stakeholders. General consumers benefit from a realistic assessment of the capital typically required to purchase a home, while governmental bodies and developers gain a credible baseline to guide large-scale housing policies and project budgets.

References

- Ministry of Land, Infrastructure, Transport and Tourism (MLIT). [Real Estate Transaction Price Information](#). Accessed March 2025.
- e-Stat. [Portal Site of Official Statistics of Japan: Housing and Land Survey](#). Accessed March 2025.