**SUBJECT ANALYTICAL PROGRAMMING**

**PROJECT 1**

**DAXIT GOLAKIYA**

**AIM ANALYSIS CONSTRUCTION DATA**

I CHOOSE THIS DATA FROM <https://catalog.data.gov/dataset>. I CHOSE THE CONSTRUCTION DATA FOR THIS SPECIFIC PYTHON ANALYSIS PROCESS. IN THIS PROJECT I WILL DEFINE NUMPY, REGEX, AND THE GRAPHICAL EXPLANATION OF THIS DATA. THIS ONE IS VERY USEFUL FOR THOSE PEOPLE WHO DO NOT HAVE KNOWLEDGE OF DETAILED CONSTRUCTION WORK ALSO THEY GET EASY KNOWLEDGE OF THE CONSTRUCTION FIELD. MY PAST IS CONSTRUCTION-RELATED SO I HAVE MORE KNOWLEDGE ABOUT THE REAL ESTATE FIELD SO I PREFER THE CONSTRUCTION DATA OVER OTHER DATA.

I MAKE DIFFERENT TYPES OF QUESTIONS AND THEIR PYTHON CODE. THIS DATA IS USEFUL FOR ECONOMIC ANALYSIS, URBAN PLANNING, REAL ESTATE MARKET, INFRASTRUCTURE INVESTMENT, POLICY DEVELOPMENT, AND COMMUNITY DEVELOPMENT.

**1. WHAT IS THE DISTRIBUTION OF BUILDING SIZES (IN SQUARE FEET) ACROSS DIFFERENT CITIES?**

**CODE**:

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Group the data by Bldg City and calculate the sum of Bldg ANSI Usable

city\_building\_sizes = df.groupby("Bldg City")["Bldg ANSI Usable"].sum()

# Display the distribution of building sizes over different cities

print(city\_building\_sizes)

**OUTPUT:**

Bldg City

ABERDEEN 209607

ABILENE 53760

ABINGDON 56049

ACCOMAC 4480

ACKERMAN 4285

...

YUBA CITY 10167

YUCCA VALLEY 4995

YUMA 159870

ZANESVILLE 9898

ZUNI 1841

THE OUTPUT IS A TABLE LISTING ALL OF THE CITIES TOGETHER WITH THE TOTAL SQUARE FOOTAGE OF USABLE SPACE IN ALL OF THEIR STRUCTURES. WE CAN EXAMINE HOW THE DISTRIBUTION OF BUILDING SIZES IS THROUGHOUT THE VARIOUS CITIES IN MY DATASET. AS STATED IN THE PREVIOUS RESPONSE, IT'S A HELPFUL SYNOPSIS FOR COMPREHENDING THE BUILT ENVIRONMENT OF DIFFERENT CITIES AND CAN BE APPLIED TO A NUMBER OF FIELDS, INCLUDING URBAN PLANNING, REAL ESTATE DEVELOPMENT, ECONOMIC RESEARCH, AND MORE.

**2. HOW MANY BUILDINGS ARE OWNED VERSUS LEASED?**

**CODE:**

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Count the number of owned and leased buildings

owned\_count = (df["Ownership Status"] == "OWNED").sum()

leased\_count = (df["Ownership Status"] == "LEASED").sum()

# Display the results

print(f"Number of owned buildings: {owned\_count}")

print(f"Number of leased buildings: {leased\_count}")

**OUTPUT:**

Number of owned buildings: 2047

Number of leased buildings: 6723

THIS OUTPUT IS HELPFUL FOR ANALYZING A DATASET OF BUILDINGS AND SPECIFICALLY EXAMINING EACH BUILDING'S OWNERSHIP STATUS. THE MAIN GOAL IS TO DETERMINE FINANCIAL ANALYSIS SINCE OWNED AND LEASED BUILDINGS MAY IMPLY DIFFERENT FINANCES. WHILE LEASED BUILDINGS MAY BE SEEN AS LIABILITIES OR EXPENSES, OWNED BUILDINGS MAY BE CONSIDERED ASSETS. THIS DATA MAY BE USEFUL FOR BUDGETING OR FINANCIAL PLANNING.

OVERALL PLANNING: COMPANY STRATEGY MAY BE AFFECTED BY THE RATE OF OWNED TO LEASED BUILDINGS. FOR INSTANCE, A COMPANY MAY CONCENTRATE ON MAKING USE OF ITS MANY BUILDINGS IF IT OWNS THEM. IT MAY SEARCH TO REDUCE LEASING COSTS OR CONSIDER BUYING MORE PROPERTIES IF IT LEASES MANY BUILDINGS. 2047 BUILDINGS ARE OWNED, AND 6723 ARE LEASED. THIS IMPLIES THAT THE BULK OF THE STRUCTURES IN THIS DATASET ARE ON LEASED.

**3. HOW MANY BUILDINGS ARE THERE IN EACH CITY?**

**CODE:**

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Group the data by Bldg City and count of buildings in each city

city\_building\_counts = df['Bldg City'].value\_counts()

# Show the number of buildings in each city

print(city\_building\_counts)

**OUTPUT:**

WASHINGTON 433

LAREDO 90

EL PASO 84

LAKEWOOD 73

ARLINGTON 70

...

CHESTERFIELD TOWNSHIP 1

WALLED LAKE 1

KINCHELOE 1

FORT GRATIOT 1

WINNEMUCCA 1

THIS CODE'S OBJECTIVE IS TO TALLY THE NUMBER OF BUILDINGS IN EVERY CITY.

THIS DATASET OUTPUT IS UDEFUL FOR THE:

URBAN PLANNING: ZONING REGULATIONS, RESOURCE ALLOCATION, INFRASTRUCTURE DEVELOPMENT, AND BUILDING DISTRIBUTION CAN ALL BE INFLUENCED BY AN UNDERSTANDING OF HOW STRUCTURES ARE DISTRIBUTED THROUGHOUT CITIES.

REAL ESTATE MARKET ANALYSIS: FOR INVESTORS, REAL ESTATE FIRMS, AND POLICY MAKERS, THE QUANTITY AND ACTIVITY OF A CITY'S BUILDING STOCK CAN PROVIDE INSIGHT INTO THE SIZE AND STATE OF THE REAL ESTATE MARKET.

DEMOGRAPHIC STUDIES: A CITY'S BUILDING COUNT CAN REVEAL INFORMATION ABOUT ITS POPULATION DENSITY, TRENDS IN URBANIZATION, AND CHANGES IN ITS DEMOGRAPHIC MAKEUP.

ACCORDING TO THIS DATA, WASHINGTON HAS THE MOST BUILDINGS (433), FOLLOWED BY ARLINGTON (70), LAKEWOOD (73), LAREDO (90), AND EL PASO (84).

**4. WHAT IS THE OLDEST AND NEWEST BUILDING IN THE DATASET? WHICH CITIES ARE THEY LOCATED IN?**

**CODE:**

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Convert the "Construction Date" column to a datetime format

df['Construction Date'] = pd.to\_datetime(df['Construction Date'], errors='coerce')

# Find the oldest building

oldest\_building = df[df['Construction Date'] == df['Construction Date'].min()]

# Find the newest building

newest\_building = df[df['Construction Date'] == df['Construction Date'].max()]

# Display the oldest and newest buildings and their respective cities

print("Oldest Building:")

print(oldest\_building[['Bldg Address1', 'Bldg City', 'Construction Date']])

print("\nNewest Building:")

print(newest\_building[['Bldg Address1', 'Bldg City', 'Construction Date']])

**OUTPUT:**

Oldest Building:

Bldg Address1 Bldg City Construction Date

1873 127 N WATER ST OGDENSBURG 1809-01-01

Newest Building:

Bldg Address1 Bldg City Construction Date

1015 999 N CAPITOL ST NE WASHINGTON 2072-11-18

THE PURPOSE OF THIS CODE IS TO IDENTIFY THE OLDEST AND NEWEST BUILDINGS IN THE DATASET AND THE CITIES THEY ARE LOCATED IN.

HISTORICAL ANALYSIS: A CITY'S ARCHITECTURAL HERITAGE AND HISTORY CAN BE CHECKED FROM ITS OLDEST STRUCTURE. HISTORIANS, ARCHITECTS, AND TOURISTS MAY FIND IT INTERESTING.

URBAN DEVELOPMENT STUDIES: A CITY'S MOST RECENT CONSTRUCTION CAN PROVIDE INSIGHT INTO ITS GROWTH PATTERNS AND RECENT URBAN DEVELOPMENT.

REAL ESTATE MARKET ANALYSIS: A BUILDING'S AGE CAN AFFECT ITS ATTRACTION TO PURCHASERS AND TENANTS, AS WELL AS ITS WORTH AND MAINTENANCE EXPENSES. REAL ESTATE COMPANIES CAN BENEFIT FROM THIS KNOWLEDGE.

BUILT ON JANUARY 1, 1809, THE OLDEST STRUCTURE IN OGDENSBURG IS SITUATED AT 127 N WATER ST. THE NEWEST STRUCTURE WAS BUILT ON NOVEMBER 18, 2072, AND IS SITUATED IN WASHINGTON, D.C. AT 999 N CAPITOL ST NE. THE AFOREMENTIONED USES OF THIS INFORMATION ARE POSSIBLE.

**5. HOW MANY BUILDINGS ARE LISTED AS ‘NATIONAL HISTORIC LANDMARK’, ‘NATIONAL REGISTER LISTED’, ‘NATIONAL REGISTER ELIGIBLE’, AND ‘NOT EVALUATED’?**

**CODE:**

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Count the number of buildings in each category

national\_historic\_landmark\_count = (df['Historical Type'] == 'National Historic Landmark').sum()

national\_register\_listed\_count = (df['Historical Type'] == 'National Register Listed').sum()

national\_register\_eligible\_count = (df['Historical Type'] == 'National Register Eligible').sum()

not\_evaluated\_count = (df['Historical Type'] == 'Not Evaluated').sum()

# Display the counts

print(f'Number of buildings listed as National Historic Landmark: {national\_historic\_landmark\_count}')

print(f'Number of buildings listed as National Register Listed: {national\_register\_listed\_count}')

print(f'Number of buildings listed as National Register Eligible: {national\_register\_eligible\_count}')

print(f'Number of buildings listed as Not Evaluated: {not\_evaluated\_count}')

**OUTPUT:**

Number of buildings listed as National Historic Landmark: 0

Number of buildings listed as National Register Listed: 0

Number of buildings listed as National Register Eligible: 0

Number of buildings listed as Not Evaluated: 0

ALL ANSWERS ARE ZERO BECAUSE IN THIS FULL DATABASE THERE IS NOT MENTIONED ANY OF THE BUILDINGS ARE NOT HISTORIC LANDMARKS. THIS DATA IS VERY USEFUL FOR TOURIST WHO ARE LOOKING FOR THIS TYPE OF BUILDING.

**6. WHAT IS THE DISTRIBUTION OF BUILDINGS BASED ON THEIR STATUS (ACTIVE, INACTIVE)?**

**CODE:**

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Group the data by Bldg Status and calculate the count of buildings in each status

building\_status\_distribution = df['Bldg Status'].value\_counts()

# Display the distribution of buildings based on their status

print(building\_status\_distribution)

**OUTPUT:**

ACTIVE 8718

EXCESS 39

DECOMMISSIONED 13

THIS DATA SET IS USEFUL FOR ASSET MANAGEMENT: MAINTAINING AND MANAGING ASSETS CAN BE AIDED BY HAVING AN UNDERSTANDING OF THE CONDITION OF BUILDINGS. WHILE RETIRED BUILDINGS MAY NEED TO BE SOLD OR PUT TO OTHER USES, ACTIVE BUILDINGS MAY NEED TO BE MAINTAINED ON A REGULAR BASIS.

STRATEGIC PLANNING: BUILDING CONDITIONS CAN HAVE AN IMPACT ON BUSINESS STRATEGY. A CORPORATION WITH A LARGE NUMBER OF OPERATIONAL BUILDINGS, FOR INSTANCE, WOULD CONCENTRATE ON MAKING USE OF THESE ASSETS, WHEREAS A COMPANY WITH A LARGE NUMBER OF DECOMMISSIONED BUILDINGS MIGHT CONSIDER SELLING OR FINDING NEW USES FOR THEM.

THERE ARE 39 EXCESS BUILDINGS, 13 RETIRED STRUCTURES, AND 8718 ACTIVE BUILDINGS. THIS IMPLIES THAT MOST OF THE BUILDINGS IN THIS DATASET ARE STILL IN USE. THIS COULD AFFECT HOW DECISIONS ARE MADE, DEPENDING ON THE SITUATION.

**7. ARE LARGER BUILDINGS MORE LIKELY TO BE OWNED OR LEASED?**

**CODE:**

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Calculate summary statistics for building size based on ownership status

owned\_mean\_size = df[df['Ownership Status'] == 'OWNED']['Bldg ANSI Usable'].mean()

leased\_mean\_size = df[df['Ownership Status'] == 'LEASED']['Bldg ANSI Usable'].mean()

owned\_median\_size = df[df['Ownership Status'] == 'OWNED']['Bldg ANSI Usable'].median()

leased\_median\_size = df[df['Ownership Status'] == 'LEASED']['Bldg ANSI Usable'].median()

# results

print("Owned Buildings - Mean Building Size:", owned\_mean\_size)

print("Leased Buildings - Mean Building Size:", leased\_mean\_size)

print("Owned Buildings - Median Building Size:", owned\_median\_size)

print("Leased Buildings - Median Building Size:", leased\_median\_size)

**OUTPUT:**

Owned Buildings - Mean Building Size: 67652.52369320957

Leased Buildings - Mean Building Size: 23831.597054886213

Owned Buildings - Median Building Size: 6978.0

Leased Buildings - Median Building Size: 8242.0

**8. TIME SERIES ANALYSIS: HOW HAS THE AVERAGE SIZE OF BUILDINGS CHANGED OVER TIME? ARE BUILDINGS GETTING LARGER OR SMALLER AS TIME PROGRESSES?**

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Convert the "Construction Date" column to a datetime format

df['Construction Date'] = pd.to\_datetime(df['Construction Date'], errors='coerce')

df['Construction Year'] = df['Construction Date'].dt.year

# Group the data by the construction year and calculate the average building size

average\_size\_by\_year = df.groupby('Construction Year')['Bldg ANSI Usable'].mean()

# Create a line chart

plt.figure(figsize=(12, 6))

average\_size\_by\_year.plot(kind='line')

plt.title('Average Building Size Over Time')

plt.xlabel('Construction Year')

plt.ylabel('Average Building Size (sq. ft)')

plt.grid(True)

plt.show()

**OIUTPUT:**

A graph showing a number of buildings

Description automatically generated

THIS CODE IS MEANT TO HELP US UNDERSTAND HOW BUILDING SIZES HAVE CHANGED OVER TIME. THIS COULD BE HELPFUL FOR SEVERAL REASONS, INCLUDING:

ARCHITECTURAL TRENDS: BUILDING AVERAGE SIZES CAN PROVIDE INSIGHT INTO CURRENT DEVELOPMENTS IN ARCHITECTURE AND DESIGN. FOR INSTANCE, IF STRUCTURES ARE GROWING LARGER OVER TIME, THIS COULD INDICATE A TREND TOWARD LARGER-THAN-LIFE DESIGNS.

URBAN PLANNING: DECISIONS ABOUT URBAN PLANNING MAY BE INFLUENCED BY THE BUILDING SIZE TREND. CITIES MAY NEED TO TAKE INTO ACCOUNT ZONING REGULATIONS TO ALLOW THE CONSTRUCTION OF LARGER STRUCTURES, FOR INSTANCE, IF BUILDING SIZES ARE INCREASING.

FROM THIS DATA CHART WE CAN CLEARLY SEE THAT OVER THE TIME THE BUILDING SIZE IS INCREASING. SO, THE POPULATION DENSITY IS GETTING HIGHER SO THE LAND COST GETTING HIGHER. WHEN IT COMES TO COMPREHENDING THE INFORMATION AND ARRIVING AT WISE CONCLUSIONS, THIS VISUAL REPRESENTATION CAN BE QUITE USEFUL.

**9. ARE THERE ANY OUTLIERS IN THE DATASET? FOR EXAMPLE, EXTREMELY LARGE OR SMALL BUILDINGS, OR BUILDINGS WITH AN UNUSUALLY HIGH NUMBER OF PEOPLE?**

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Define a threshold

threshold = 2

# Calculate the Z-scores

z\_scores = (df['Bldg ANSI Usable'] - df['Bldg ANSI Usable'].mean()) / df['Bldg ANSI Usable'].std()

# Identify potential outliers

outliers = df[abs(z\_scores) > threshold]

# Calculate the proportion of buildings with potential outliers

total\_buildings = len(df)

outlier\_buildings = len(outliers)

proportion\_outliers = outlier\_buildings / total\_buildings

# Create a pie chart

sizes = [proportion\_outliers, 1 - proportion\_outliers]

colors = ['red', 'lightblue']

explode = (0.1, 0)

plt.pie(sizes, labels=labels, colors=colors, explode=explode, autopct='%1.1f%%', startangle=140)

plt.title('Proportion of Buildings with Potential Outliers')

plt.axis('equal')

plt.show()

**OUTPUT:**

A blue circle with a red triangle

Description automatically generated

THIS CODE'S OBJECTIVE IS TO FIND ANY STRUCTURES THAT ARE ABNORMALLY HUGE OR SMALL—BUILDINGS THAT ARE REGARDED AS OUTLIERS.

QUALITY CONTROL: ERRORS IN THE DATA ENTRY OR COLLECTION MAY OCCASIONALLY BE INDICATED BY OUTLIERS. FINDING THESE CAN ENHANCE THE DATASET'S CORRECTNESS AND DEPENDABILITY.

RISK MANAGEMENT: DIFFERENT RISKS MAY ARISE FROM EXTREMELY LARGE OR SMALL BUILDINGS (E.G., LARGER BUILDINGS MAY REQUIRE MORE CARE, OR SMALLER BUILDINGS MAY BE LESS USEFUL). FINDING THESE CAN AID IN MANAGING AND ASSESSING RISKS.

**10. WHAT IS THE AVERAGE SIZE OF BUILDINGS IN EACH CITY?**

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Group the data by Bldg City and calculate the average of Bldg ANSI Usable

city\_avg\_building\_size = df.groupby("Bldg City")["Bldg ANSI Usable"].mean()

# Create a bar chart to visualize the average building size in each city

city\_avg\_building\_size.plot(kind='bar', figsize=(12, 6))

plt.title('Average Building Size by City')

plt.xlabel('City')

plt.ylabel('Average Building Size (sq. ft)')

plt.xticks(rotation=90) # Rotate x-axis labels for better read

plt.show()

**OUTPUT:**

A graph of a building size

Description automatically generated with medium confidence

THIS CODE'S OBJECTIVE IS TO DETERMINE AND DISPLAY EACH CITY'S AVERAGE BUILDING SIZE. THIS COULD BE HELPFUL FOR SEVERAL REASONS, INCLUDING:

URBAN PLANNING: BY KNOWING THE TYPICAL BUILDING SIZE, PLANNERS MAY BETTER ALLOCATE RESOURCES, DESIGN INFRASTRUCTURE, AND COMPLY WITH ZONING REGULATIONS.

REAL ESTATE INDUSTRY ANALYSIS: FOR INVESTORS, REAL ESTATE FIRMS, AND POLICY MAKERS, THE AVERAGE BUILDING SIZE IN A CITY CAN PROVIDE INSIGHT INTO THE NATURE AND MAGNITUDE OF THE REAL ESTATE INDUSTRY.

THIS CHART IT IS SHOWS THAT SOME CITIES HAVE LARGER BUILDINGS IN SQUARE FEET WISE. IF WE SELECT A SMALL DATA BASE, WE CAN CLEARLY SEE THE PROPER CITY AND THEIR BUILDING SIZE.

**11. USING NUMPY, FIND THE MEDIAN OF THE TOTAL PARKING SPACES IN THE DATASET.**

**CODE:**

import numpy as np

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Extract the 'Total Parking Spaces.

total\_parking\_spaces = df['Total Parking Spaces'].values

# Calculate the median using numpy

median\_parking\_spaces = np.median(total\_parking\_spaces)

print("Median of Total Parking Spaces:", median\_parking\_spaces)

**OUTPUT:**

Median of Total Parking Spaces: 8.0

THIS CODE'S GOAL IS TO DETERMINE THE MEDIAN AMOUNT OF PARKING SPACES OVERALL FOR EACH BUILDING IN THE DATASET.

TRANSPORTATION STUDIES: INFORMATION ON PARKING SPACES' MEDIAN QUANTITY CAN ALSO REVEAL PATTERNS AND REQUIREMENTS IN THE FIELD OF TRANSPORTATION.

THERE ARE 8.0 PARKING SPOTS OVERALL, WHICH IS THE MEDIAN. THIS INDICATES THAT 8.0 IS THE MIDDLE FIGURE WHEN ALL THE BUILDINGS ARE ARRANGED ACCORDING TO THE TOTAL NUMBER OF PARKING SPOTS. THE AFOREMENTIONED USES OF THIS INFORMATION ARE POSSIBLE.

**12. USE REGEX TO EXTRACT THE CITY NAMES FROM THE BUILDING ADDRESSES.**

**CODE:**

import re

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Define a regular expression pattern with a capture group to match city names

city\_pattern = r'([^,]+)(?=\s+[A-Z]{2}\s+\d{5}$)'

# Extract city names using the regular expression and create a new column

df['City'] = df['Bldg Address1'].str.extract(city\_pattern)

# Now, the 'City' column contains the extracted city names

print(df['City'])

**OUTPUT:**

0 NaN

1 NaN

2 NaN

3 NaN

4 NaN

...

8765 NaN

8766 NaN

8767 NaN

8768 NaN

8769 NaN

THIS CODE USES A REGULAR EXPRESSION (REGEX) TO EXTRACT THE CITY NAMES FROM THE BUILDING ADDRESSES.

CLEANING DATA: CLEANING AND STANDARDIZING THE DATA CAN BE AIDED BY EXTRACTING CITY NAMES. THE DATA ANALYSIS BY CITY MAY BECOME SIMPLER AS A RESULT.

GEOGRAPHICAL ANALYSIS: CITY NAMES CAN GIVE THE DATA A GEOGRAPHICAL CONTEXT THAT IS HELPFUL FOR A VARIETY OF ANALYSES, SUCH AS REAL ESTATE MARKET ANALYSIS AND URBAN PLANNING.

DATA ENRICHMENT: BY ADDING NEW INFORMATION TO THE DATASET THAT CAN BE UTILIZED FOR ANALYSIS, THE EXTRACTION OF CITY NAMES CAN IMPROVE ITS QUALITY.

**13. FIND ALL BUILDING ADDRESSES THAT INCLUDE 'U.S. COURTHOUSE' IN THEIR ADDRESS LINE.**

**CODE:**

import re

import pandas as pd

# Read the data from the CSV file

df = pd.read\_csv('Construction Data.csv')

# Find addresses containing "U.S. COURTHOUSE"

pattern = r'.\*U\.S\. COURTHOUSE.\*'

us\_courthouse\_addresses = df[df['Bldg Address1'].str.contains(pattern, case=False)]

# Now, `us\_courthouse\_addresses` contains all rows with "U.S. COURTHOUSE" in their address

print(us\_courthouse\_addresses)

OUTPUT:

Empty DataFrame

LEGAL STUDIES: THIS CODE CAN ASSIST US IN IDENTIFYING EVERY COURTHOUSE IN THE DATASET IF WE ARE RESEARCHING HOW COURTHOUSES ARE DISTRIBUTED ACROSS THE NATION.