Task 04: Extract descriptive statistics from a housing prices and Iris datasets.

Objective: Use Python's libraries such as pandas, numpy, scipy, seaborn and matplotlib to perform statistical manipulations and visualize the results.

Instructions:

Step 1: Load the Dataset

1. Load the **Housing Prices dataset** into a Pandas DataFrame.
2. Display the first few rows of the dataset to verify it has been loaded correctly.

Step 2: Data Exploration

1. Check the structure of the dataset by printing its shape (number of rows and columns) and the column names. How many features do the dataset contain?
2. Display summary statistics (e.g., mean, median, quartiles) of the numerical features. What is the average petal length for each species?
3. Check if there are any missing values in the dataset.

Step 3: Extract Descriptive Statistics

1. Find the basic statistical measures such as mean, median, variance and standard deviation for the housing prices.
2. Find the quartiles (Q1, Q2 and Q3) and evaluate the interquartile range. Use the 1.5*IQR rule to find the outliers in the data.
3. Use boxplot to visualize the outliers.
4. Use the scipy.stats to estimate the pdf and cdf of the housing prices.
5. The "Neighborhood" column is categorical and can be encoded as Suburbs =1, Rural = 2 and Urban = 3. Assign these numerical values to the "Neighborhood" by replacing the categorical ones.
6. To understand how the features are correlated with the price, create a correlation matrix and visualize using a heatmap.

Step 4: Data Manipulation

1. Perform data Normalization using Min-Max scaling.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

```
from sklearn.preprocessing import MinMaxScaler


# Define the scaler

scaler = MinMaxScaler()

# Select features to normalize (excluding target 'price')

features = ['square_feet', 'no_of_bedrooms', 'no_of_bathrooms', 'year_built']  # Include any relevant features

df[features] = scaler.fit_transform(df[features])
```

2. Perform data Normalization using Z-score Normalization by transforming the data to have a mean of 0 and a standard deviation of 1.

$$X_{standardized} = \frac{X - \mu}{\sigma}$$

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
```

3. Use Robust Normalization, similar to Z-score normalization but uses the median and the interquartile range (IQR), making it robust to outliers.

$$X_{robust} = \frac{X - median}{IQR}$$

```
from sklearn.preprocessing import RobustScaler

scaler =RobustScaler()
```

4. Visualize how the normalization impacts the feature distributions by using histograms and boxplots before and after each normalization.

```
import matplotlib.pyplot as plt


# Plot histograms of the features before normalization
df_orig = pd.read_csv('housing_prices.csv')
df_orig[features].hist(bins=20, figsize=(10, 8))
plt.suptitle("Before Normalization")
plt.show()


# Plot histograms of the normalized features
df[features].hist(bins=20, figsize=(10, 8))
plt.suptitle("After Normalization")
plt.show()
```

Step 5: Perform all these tasks on Iris Dataset.