Name : Jainil Soni

Student Id : 202412107

Assignment : 2

Collab Notebook Link : https://colab.research.google.com/drive/1SuqHGCF1wusfG1Up9l_d5wnU38WDasVK?usp=sharing

Assignment Solution

Question 1 : Advanced Data Filtering (Pandas)(10 marks)

Find and display all the housing blocks that are less than 15 years old (HouseAge < 15) and have an average number of rooms (AveRooms) greater than 6. How many such housing blocks exist in the dataset? Display the first 5 rows of the filtered data.

```
import pandas as pd
from sklearn.datasets import fetch_california_housing
import matplotlib.pyplot as plt
import seaborn as sns

california_housing = fetch_california_housing()

df = pd.DataFrame(california_housing.data, columns=california_housing.feature_names)
df['MedHouseValue'] = california_housing.target

mask = (df['HouseAge'] < 15) & (df['AveRooms'] > 6)

filtered_df = df.loc[mask].copy()

count_blocks = filtered_df.shape[0]

print(f"Number of housing blocks with HouseAge < 15 and AveRooms > 6: {count_blocks}")

filtered_df.head(5)
```

```
Number of housing blocks with HouseAge < 15 and AveRooms > 6: 1199
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseValue
570	7.6110	5.0	6.855776	1.061442	7427.0	2.732524	37.72	-122.24	3.507
576	7.2634	12.0	7.133034	1.018934	5781.0	2.880419	37.77	-122.06	3.416
577	7.0568	5.0	7.023438	0.912109	1738.0	3.394531	37.73	-122.06	4.125
706	6.2579	10.0	6.443323	1.029503	3827.0	2.971273	37.65	-122.04	3.155
838	5.0406	6.0	6.016166	1.094688	1568.0	3.621247	37.61	-122.08	2.614

Next steps: Generate code with filtered_df | View recommended plots | New interactive sheet

Step by step explanation of the code :

Loads data & builds a DataFrame : Uses fetch_california_housing() and puts features into df, then adds the target column as MedHouseValue.

Creates a filter (mask) : (df['HouseAge'] < 15) & (df['AveRooms'] > 6) produces a True/False Series for rows meeting both conditions.

Applies the filter : df.loc[mask] returns only rows where the mask is True; .copy() avoids chained-assignment warnings.

Counts matches : filtered_df.shape[0] gives the number of rows that match the criteria.

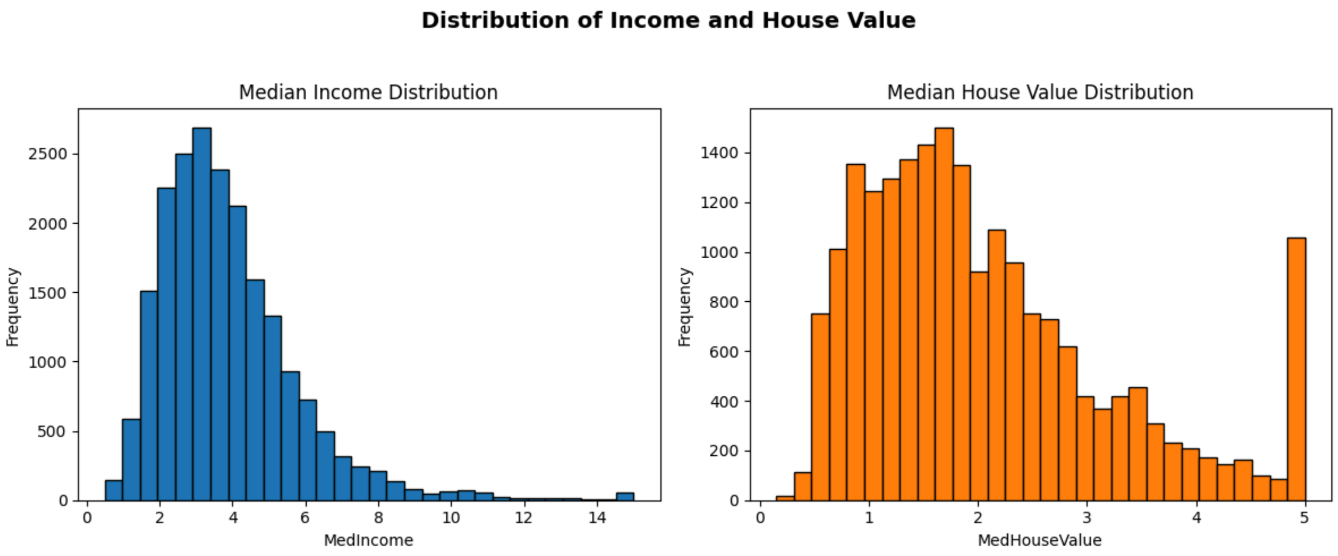
Shows sample rows : filtered_df.head(5) prints the first five matching rows, as requested.

Question 3 : Comparative Analysis with Subplots (Matplotlib)(20 marks)

Create a figure with two subplots arranged side-by-side (1 row, 2 columns).

1. In the left subplot, plot a histogram of the MedInc (Median Income).
2. In the right subplot, plot a histogram of the MedHouseValue (Median House Value). Customize your plots: ● Set a unique title for each subplot. ● Label the x-axis for both plots. ● Give the entire figure a main title, for example, "Distribution of Income and House Value". ● Use a different color for each histogram.

```
import matplotlib.pyplot as plt

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))

axes[0].hist(df['MedInc'], bins=30, color='tab:blue', edgecolor='black')
axes[0].set_title('Median Income Distribution')
axes[0].set_xlabel('MedIncome')
axes[0].set_ylabel('Frequency')

axes[1].hist(df['MedHouseValue'], bins=30, color='tab:orange', edgecolor='black')
axes[1].set_title('Median House Value Distribution')
axes[1].set_xlabel('MedHouseValue')
axes[1].set_ylabel('Frequency')

fig.suptitle('Distribution of Income and House Value', fontsize=14, fontweight='bold')

fig.tight_layout(rect=[0, 0, 1, 0.95])

plt.show()
```



Step by step explanation of the code :

Create subplots : plt.subplots(1, 2, figsize=(12,5)) makes a figure with two side-by-side axes.

Left histogram : Plots df['MedInc'] with 30 bins, sets a unique title and x-label.

Right histogram : Plots df['MedHouseValue'] similarly, with a different color and its own title/x-label.

Main title : fig.suptitle(...) adds a figure-level title.

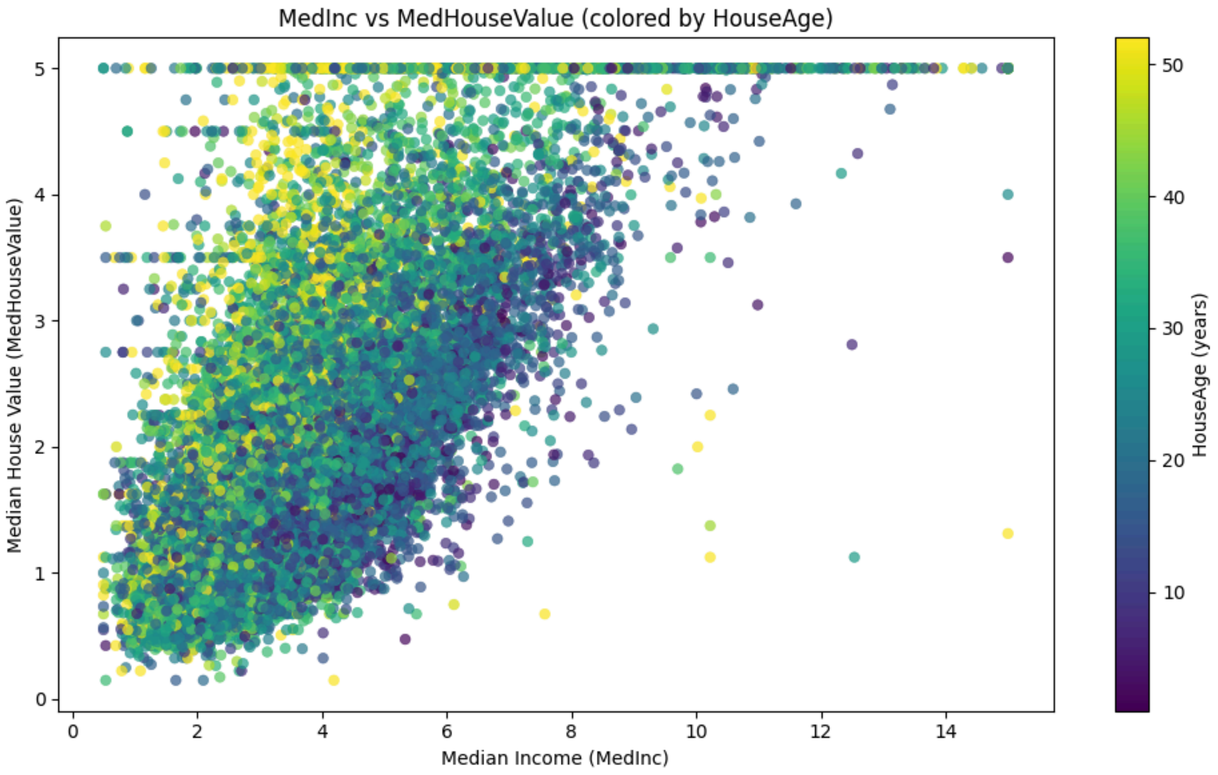
Layout : tight_layout(rect=[0,0,1,0.95]) keeps the suptitle visible and prevents overlap.

Render : plt.show() displays the figure.

Question 4 : Exploring Relationships (Seaborn)(20 marks)

Visualize the relationship between Median Income (MedInc) and Median House Value (MedHouseValue). ● Create a scatter plot using Seaborn. ● Use the HouseAge of the property to color the points on the scatter plot. A continuous color scale should be used. ● What does the plot tell you about the relationship between income, house value, and the age of the house? Provide a brief one-sentence interpretation.

```
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.colors import Normalize

fig, ax = plt.subplots(figsize=(10, 6))

norm = Normalize(vmin=df['HouseAge'].min(), vmax=df['HouseAge'].max())

scatter = sns.scatterplot(
scatter = sns.scatterplot(
    data=df,
    x='MedInc',
    y='MedHouseValue',
    hue='HouseAge',
    palette='viridis',
    hue_norm=norm,
    s=30,
    alpha=0.7,
    edgecolor=None,
    ax=ax
)

legend = scatter.get_legend()

if legend:
    legend.remove()

sm = plt.cm.ScalarMappable(cmap='viridis', norm=norm)
sm.set_array([])

cbar = fig.colorbar(sm, ax=ax)
cbar.set_label('HouseAge (years)')

ax.set_title('MedInc vs MedHouseValue (colored by HouseAge)')
ax.set_xlabel('Median Income (MedInc)')
ax.set_ylabel('Median House Value (MedHouseValue)')

plt.tight_layout()
plt.show()
```



Step by step code explanation :

Imports : Bring in matplotlib, seaborn, and Normalize to map numeric ages to the colormap.

Figure/axis : Create a single plotting axis with a comfortable figure size.

Normalize : Build a Normalize object from the min/max of HouseAge so the color mapping is scaled correctly.

Plot with Seaborn : Use sns.scatterplot(...) with hue='HouseAge', palette='viridis', and hue_norm=norm so point color continuously reflects house age; alpha and s help readability.

Remove the default legend : Seaborn's discrete legend isn't suitable for continuous hues, so remove it.

Add colorbar : Create a ScalarMappable (same cmap + norm) and attach a colorbar to the figure so users can read the HouseAge values from the colors.

Labels & show : Set title and axis labels, tighten layout, and display the plot.

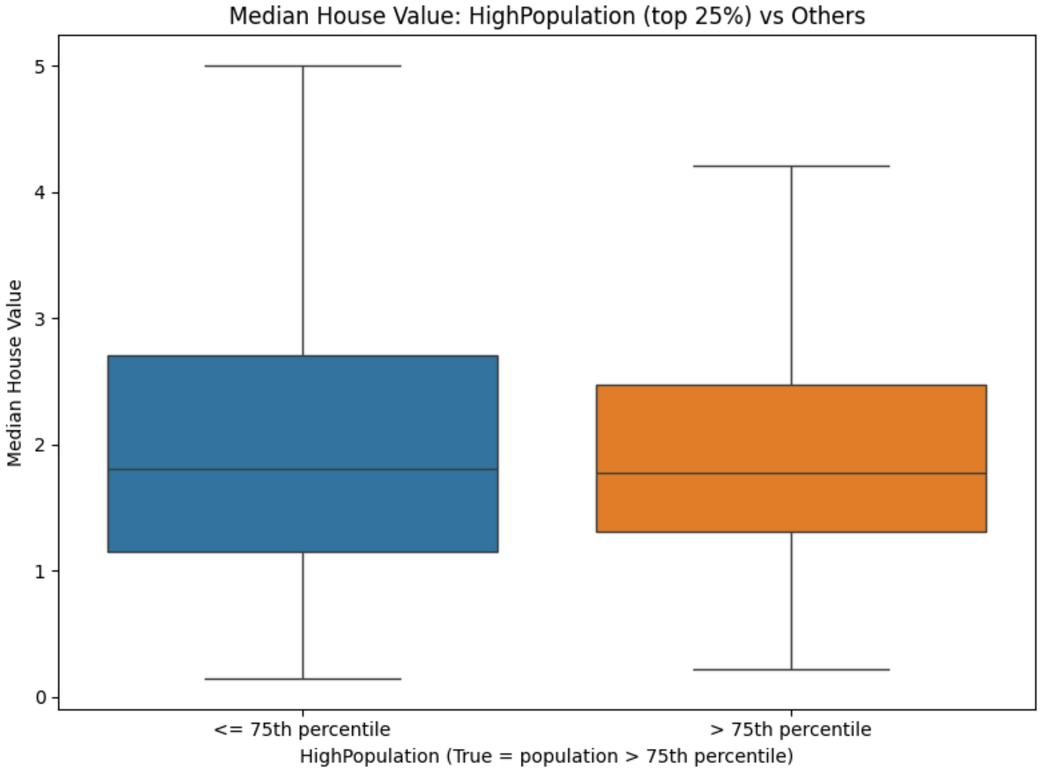
One sentence interpretation(Conclusion) : There is a clear positive trend between median income and median house value (higher incomes generally correspond to higher house values); the HouseAge color gradient appears mixed across the cloud, suggesting house age does not solely determine value — both newer and older homes can be found at different value levels

Question 5 : Combining Manipulation and Visualization (Pandas & Seaborn)(30 marks)

Identify the housing blocks with the highest population density.

1. (Pandas) Create a new boolean column named HighPopulation. This column should be True if the Population is greater than the 75th percentile of the population and False otherwise.
2. (Seaborn & Matplotlib) Use a Seaborn boxplot to compare the MedHouseValue for the HighPopulation (True) and non-HighPopulation (False) groups.
3. (Matplotlib) Give your plot a clear title and y-axis label.
4. Based on your plot, do areas with higher population tend to have higher or lower median house values?

```
import matplotlib.pyplot as plt
import seaborn as sns

pop_75 = df['Population'].quantile(0.75)

df['HighPopulation'] = df['Population'] > pop_75

plt.figure(figsize=(8, 6))
sns.boxplot(
    x='HighPopulation',
    y='MedHouseValue',
    hue='HighPopulation',
    data=df,
    legend=False,
    showfliers=False,
    palette=['tab:blue', 'tab:orange']
)

plt.title('Median House Value: HighPopulation (top 25%) vs Others')
plt.xlabel('HighPopulation (True = population > 75th percentile)')
plt.ylabel('Median House Value')
plt.xticks([0, 1], ['<= 75th percentile', '> 75th percentile'])

plt.tight_layout()
plt.show()
```



Step by step code explanation :

Compute threshold : df['Population'].quantile(0.75) finds the population value at the 75th percentile.

Make boolean column : df['HighPopulation'] = df['Population'] > pop_75 marks rows in the top 25% of population as True. (Question required strictly greater than the 75th percentile.)

Inspect groups : value_counts() shows how many blocks are True/False; groupby(...).median() prints the median MedHouseValue for each group so you can compare numerically.

Visualize : sns.boxplot(...) draws side-by-side boxes (notches/outliers optional) for MedHouseValue for the two groups; order guarantees the box order, palette gives different colors, and showfliers=False hides extreme outliers for readability.

Label & show : add a clear title and y-axis label, replace x-tick text with descriptive labels, and display the figure.

If the median MedHouseValue for HighPopulation == True is greater than for False, then higher-population blocks tend to have higher median house values. If the median is lower, then higher-population blocks tend to have lower median house values. The boxplot gives the distribution (spread, skew, and outliers) while the printed medians give a precise central-value comparison.

Double-click (or enter) to edit