Atharva Yadav
T23 - 101
DSPL Assg 5

1. Calculate covariance(Age, Strength) and correlation(Age, strength) for following data using a. Pearson's Correlation coefficient method b. Spearman's rank correlation method (use scipy.stats.pearsonr and scipy.stats.spearmanr functions) Age 38 62 22 38 45 69 75 38 80 32 51 56 21 34 76 Strength 20 15 30 21 18 12 14 28 09 22 20 19 28 23 14

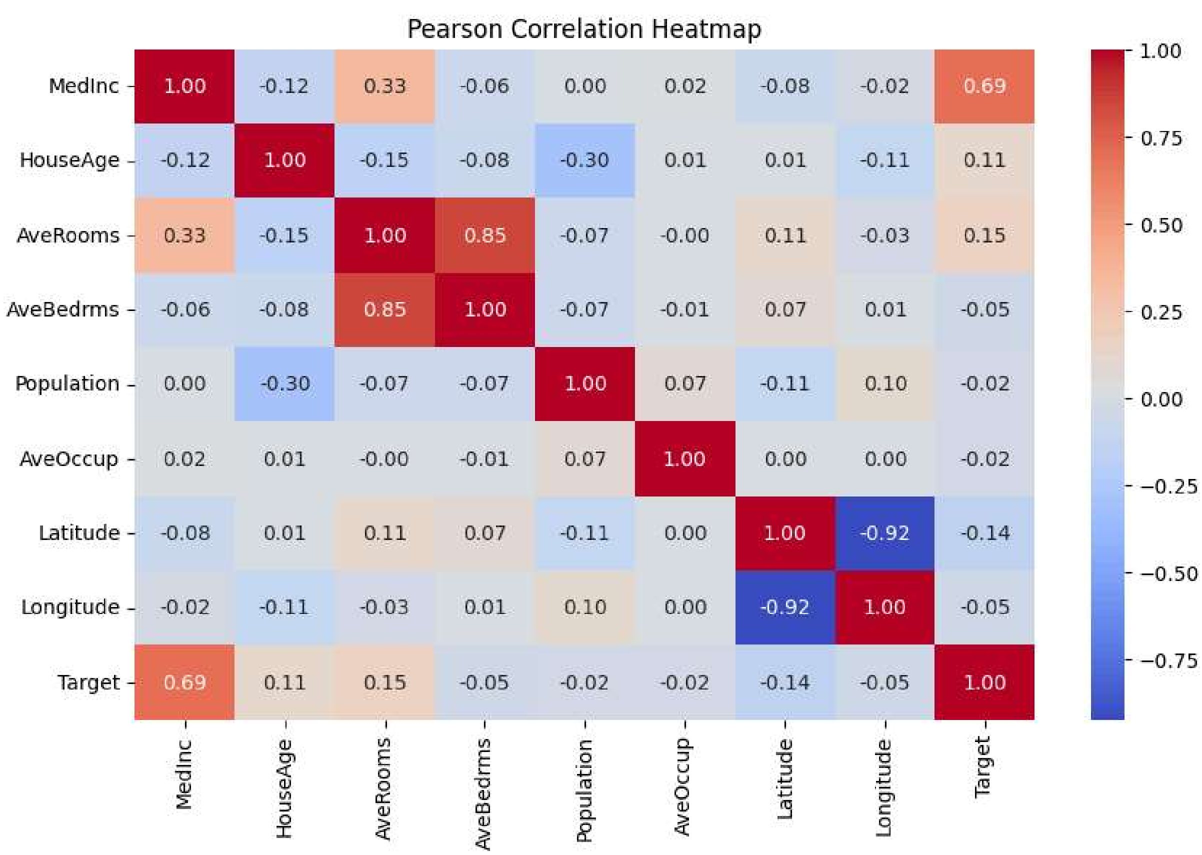
In [2]:
```
import numpy as np
import scipy.stats as stats

# Given data
age = np.array([38, 62, 22, 38, 45, 69, 75, 38, 80, 32, 51, 56, 21, 34, 76])
strength = np.array([20, 15, 30, 21, 18, 12, 14, 28, 9, 22, 20, 19, 28, 23, 14])

# Calculate covariance
covariance_matrix = np.cov(age, strength)
covariance = covariance_matrix[0, 1]  # Extract covariance value

# Calculate Pearson's correlation coefficient
pearson_corr, pearson_p_value = stats.pearsonr(age, strength)

# Calculate Spearman's rank correlation coefficient
spearman_corr, spearman_p_value = stats.spearmanr(age, strength)

# Display results
print(f"Covariance between Age and Strength: {covariance:.2f}")
print(f"Pearson's Correlation Coefficient: {pearson_corr:.2f}, p-value: {pearson_p_
print(f"Spearman's Rank Correlation Coefficient: {spearman_corr:.2f}, p-value: {spe
```

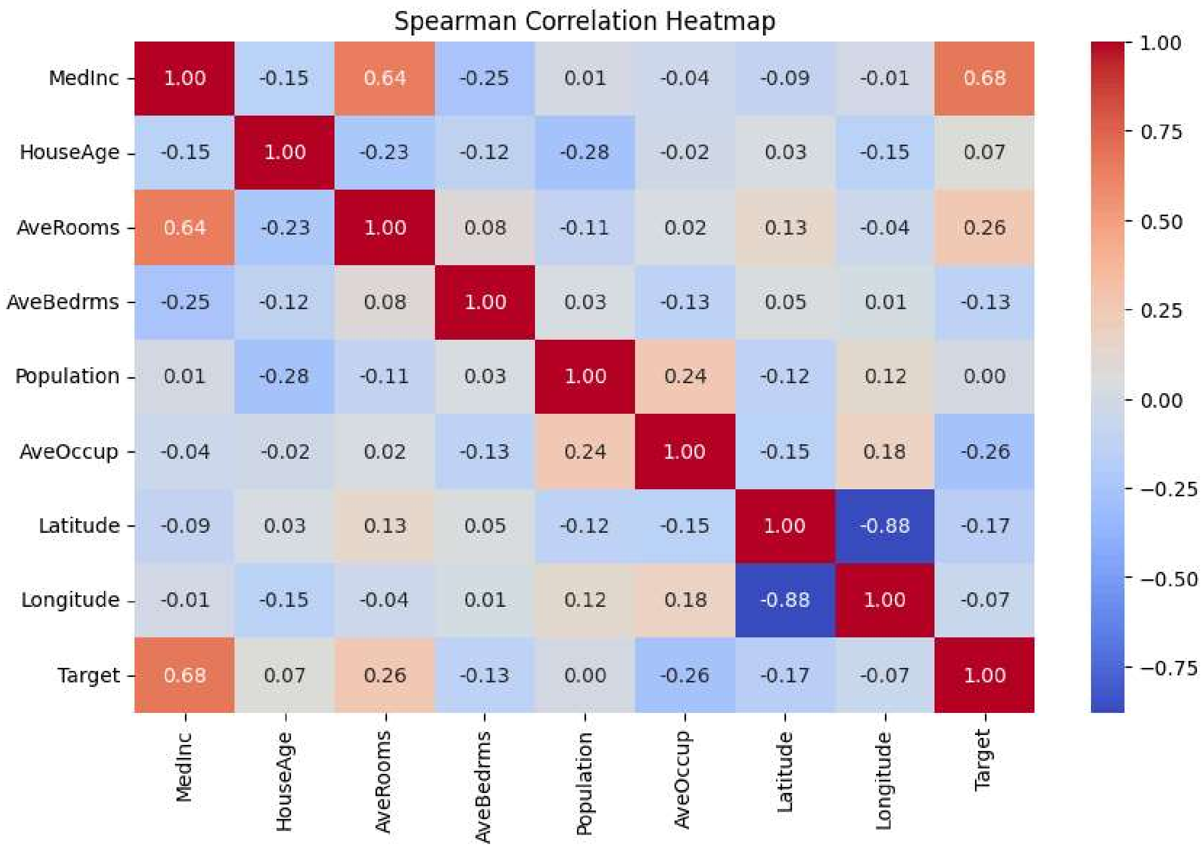
```
Covariance between Age and Strength: -111.00
Pearson's Correlation Coefficient: -0.92, p-value: 0.0000
Spearman's Rank Correlation Coefficient: -0.94, p-value: 0.0000
```

2. Load the california_housing dataset from scikitlearn. Find the correlation between each of the features using pearson's method and spearman's method. Draw the heatmap for both cases. Identify if any redundant features are there and which independent features are highly correlated with dependent feature. Use following lines of code to import the dataset from sklearn and then create the dataframe from it: from sklearn.datasets import fetch_california_housing housing = fetch_california_housing()

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing
import scipy.stats as stats

# Load the California Housing dataset
housing = fetch_california_housing()
df = pd.DataFrame(housing.data, columns=housing.feature_names)

# Add the target (house prices) to the DataFrame
```

```
df["Target"] = housing.target

# Compute Pearson's correlation matrix
pearson_corr = df.corr(method="pearson")

# Compute Spearman's correlation matrix
spearman_corr = df.corr(method="spearman")

# Plot Pearson's correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(pearson_corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Pearson Correlation Heatmap")
plt.show()

# Plot Spearman's correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(spearman_corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Spearman Correlation Heatmap")
plt.show()

# Identifying redundant features (highly correlated independent features)
redundant_features = pearson_corr[(pearson_corr > 0.75) & (pearson_corr < 1)].dropn
print("Highly Correlated Features (Redundant Features):")
print(redundant_features)

# Identifying independent features highly correlated with the target
target_corr = pearson_corr["Target"].sort_values(ascending=False)
print("\nFeatures Highly Correlated with Target (Dependent Feature):")
print(target_corr)
```






```
Highly Correlated Features (Redundant Features):
           AveRooms  AveBedrms
AveRooms        NaN   0.847621
AveBedrms  0.847621        NaN

Features Highly Correlated with Target (Dependent Feature):
Target        1.000000
MedInc        0.688075
AveRooms      0.151948
HouseAge      0.105623
AveOccup     -0.023737
Population   -0.024650
Longitude    -0.045967
AveBedrms    -0.046701
Latitude     -0.144160
Name: Target, dtype: float64
```