```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import geopandas as gpd
import folium

# Example: Analyzing Health Equity and Identifying Disparities in Healthcare Access

# Step 1: Load the Data
# Dataset contains the following columns:
# 'Region': Geographic region name
# 'Population': Population size of the region
# 'Healthcare_Facilities': Number of healthcare facilities in the region
# 'Median_Income': Median income of the population
# 'Disease_Prevalence': Percentage of population with a specific disease
# 'Uninsured_Rate': Percentage of population without health insurance

data = {
  'Region': ['Region A', 'Region B', 'Region C', 'Region D', 'Region E'],
  'Population': [50000, 30000, 40000, 60000, 25000],
  'Healthcare_Facilities': [5, 3, 4, 6, 2],
  'Median_Income': [55000, 35000, 45000, 65000, 30000],
  'Disease_Prevalence': [15, 25, 20, 10, 30],
```

```python
    'Uninsured_Rate': [10, 20, 15, 5, 25]
}

# Convert data to a DataFrame
df = pd.DataFrame(data)

# Step 2: Calculate Access-to-Care Metrics
df['Facilities_Per_Capita'] = df['Healthcare_Facilities'] / df['Population']
df['Disease_Burden'] = df['Disease_Prevalence'] * df['Population'] / 100

# Step 3: Exploratory Data Analysis
print("Summary Statistics:")
print(df.describe())

# Correlation Heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()

# Step 4: Clustering to Identify Underserved Regions
# Prepare data for clustering
features = ['Facilities_Per_Capita', 'Median_Income', 'Uninsured_Rate',
'Disease_Prevalence']
scaler = StandardScaler()
X = scaler.fit_transform(df[features])
```

```
# Apply K-Means Clustering
kmeans = KMeans(n_clusters=2, random_state=42)
df['Cluster'] = kmeans.fit_predict(X)

# Visualize Clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df['Median_Income'], y=df['Facilities_Per_Capita'], hue=df['Cluster'],
palette='viridis')
plt.title("Clusters of Regions Based on Healthcare Equity")
plt.xlabel("Median Income")
plt.ylabel("Facilities Per Capita")
plt.show()

# Step 5: Geospatial Analysis (Mapping Underserved Regions)
# Assume we have a shapefile of the regions (mock data)
region_geo = gpd.GeoDataFrame({
    'Region': ['Region A', 'Region B', 'Region C', 'Region D', 'Region E'],
    'geometry': gpd.points_from_xy([0, 1, 2, 3, 4], [0, 1, 2, 3, 4])  # Mock coordinates
})
region_geo = region_geo.merge(df, on='Region')

# Create an interactive map
m = folium.Map(location=[2, 2], zoom_start=6)
for _, row in region_geo.iterrows():
    folium.CircleMarker(
```

```
        location=(row.geometry.y, row.geometry.x),
        radius=10,
        color='red' if row['Cluster'] == 1 else 'blue',
        fill=True,
        fill_opacity=0.6,
        popup=f"Region: {row['Region']}<br>Facilities Per Capita:
{row['Facilities_Per_Capita']:.2f}<br>Cluster: {row['Cluster']}"
    ).add_to(m)

m.save("health_equity_map.html")

# Step 6: Insights and Recommendations
# Highlight regions in Cluster 1 (potentially underserved)
underserved = df[df['Cluster'] == 1]
print("Underserved Regions:")
print(underserved[['Region', 'Population', 'Facilities_Per_Capita', 'Median_Income',
'Uninsured_Rate']])

# Example Insight:
# Regions in Cluster 1 tend to have lower median incomes, higher uninsured rates, and
fewer facilities per capita.
# Recommendation: Prioritize these regions for healthcare resource allocation and policy
interventions.

# Save results for reporting
df.to_csv("health_equity_analysis_results.csv", index=False)
```