Healthcare Scenario Healthy Living and Wellness Clustering Exercise

Healthcare Scenario: Healthy Living and Wellness Clustering Exercise

1. Import Libraries

```
# Install these packages if you don't have them installed
# !pip install pandas seaborn matplotlib scikit-learn

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
```

2. Load Dataset

```
# Load your dataset
df = pd.read_csv(r'C:\Users\Saba\Documents\Semester - 04\Itauma\Directories\Machine_Learning\

# Show the first few rows
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
```

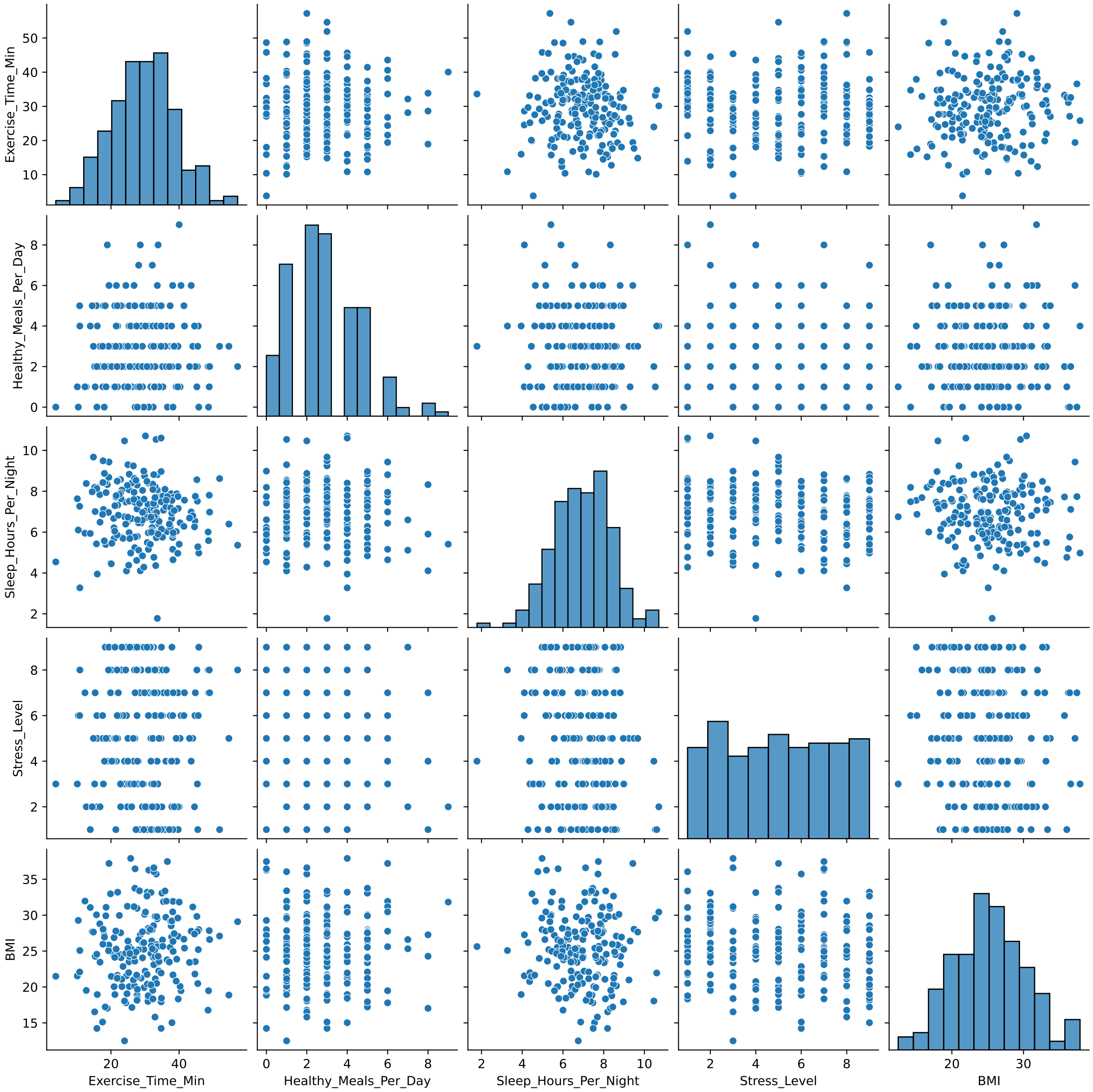
```
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Exercise_Time_Min     200 non-null    float64
 1   Healthy_Meals_Per_Day  200 non-null    int64  
 2   Sleep_Hours_Per_Night  200 non-null    float64
 3   Stress_Level          200 non-null    int64  
 4   BMI                   200 non-null    float64
dtypes: float64(3), int64(2)
memory usage: 7.9 KB
None
       Exercise_Time_Min  Healthy_Meals_Per_Day  Sleep_Hours_Per_Night  \
count         200.000000             200.000000             200.000000   
mean           29.592290               2.875000               6.933582   
std             9.310039               1.815449               1.422471   
min             3.802549               0.000000               1.778787   
25%            22.948723               2.000000               5.967243   
50%            29.958081               3.000000               6.972331   
75%            35.008525               4.000000               7.886509   
max            57.201692               9.000000              10.708419   

       Stress_Level         BMI  
count    200.000000  200.000000  
mean       4.995000   25.150008  
std        2.605556    5.070778  
min        1.000000   12.502971  
25%        3.000000   21.458196  
50%        5.000000   25.155662  
75%        7.000000   28.011155  
max        9.000000   37.898547  
```

3. Exploratory Data Analysis (EDA)

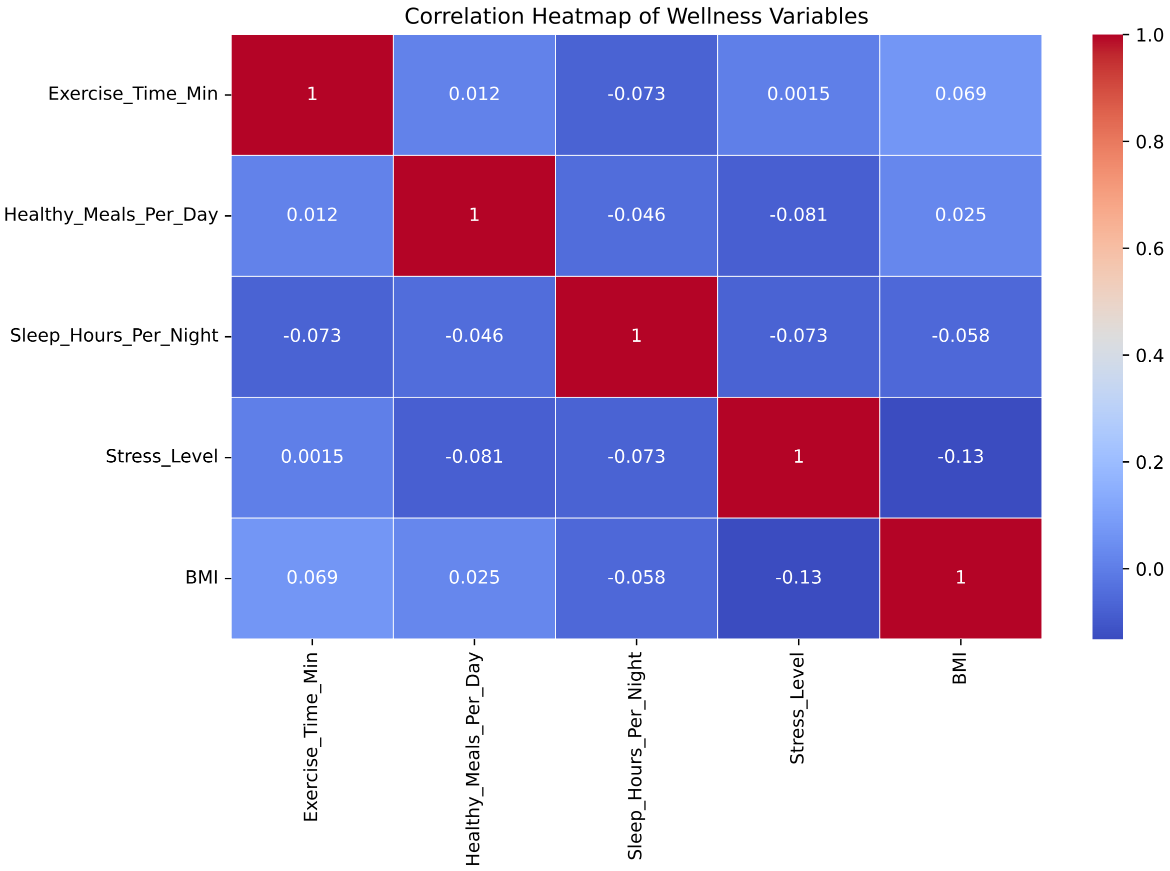
Pairplot to Visualize Relationships

```
sns.pairplot(df)
plt.show()
```



Correlation Heatmap

```
plt.figure(figsize=(10, 6))
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Heatmap of Wellness Variables")
plt.show()
```



4. Data Preprocessing

```
# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
```

5. Clustering - K-Means

```
# Fit KMeans with 3 clusters (adjust based on data exploration)
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_labels = kmeans.fit_predict(scaled_data)

# Add cluster labels to the original data
df['KMeans_Cluster'] = kmeans_labels
```

```
# Silhouette Score to measure the clustering quality
silhouette_avg = silhouette_score(scaled_data, kmeans_labels)
print(f'Silhouette Score: {silhouette_avg}')
```

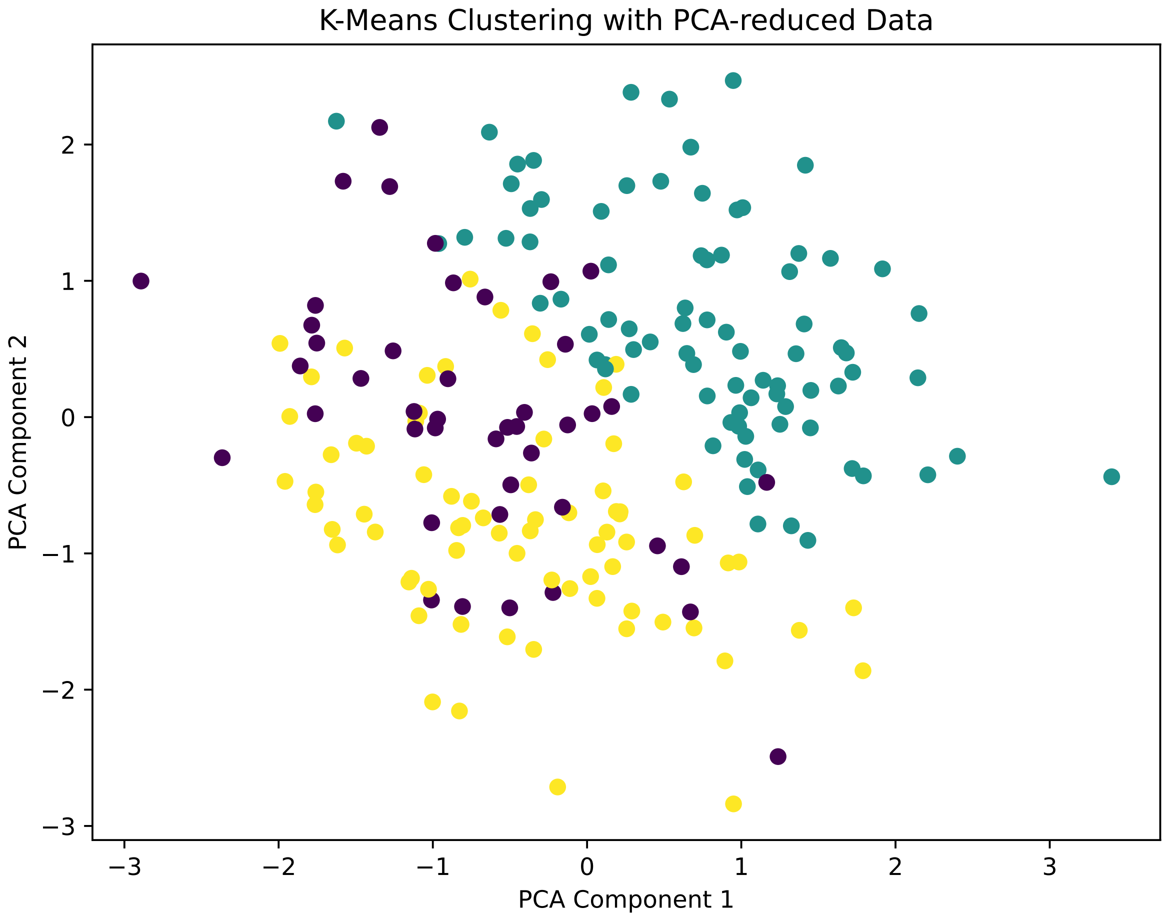
```
Silhouette Score: 0.1516159911787657
```

6. Dimensionality Reduction - PCA

```
# Apply PCA
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

# Visualize the PCA-reduced data with the K-Means clusters
plt.figure(figsize=(8, 6))
plt.scatter(pca_data[:, 0], pca_data[:, 1], c=kmeans_labels, cmap='viridis')
plt.title("K-Means Clustering with PCA-reduced Data")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()

# Check explained variance of the components
print(f'Explained Variance Ratio: {pca.explained_variance_ratio_}')
```



```
Explained Variance Ratio: [0.23691549 0.22082517]
```

7. Hierarchical Clustering

```
# Apply Agglomerative (Hierarchical) Clustering
agg_clustering = AgglomerativeClustering(n_clusters=3)
agg_labels = agg_clustering.fit_predict(scaled_data)

# Add hierarchical cluster labels to the dataset
df['Agg_Cluster'] = agg_labels

# Compare silhouette scores
silhouette_avg_agg = silhouette_score(scaled_data, agg_labels)
print(f'Agglomerative Clustering Silhouette Score: {silhouette_avg_agg}')
```

```
Agglomerative Clustering Silhouette Score: 0.13628495765267165
```

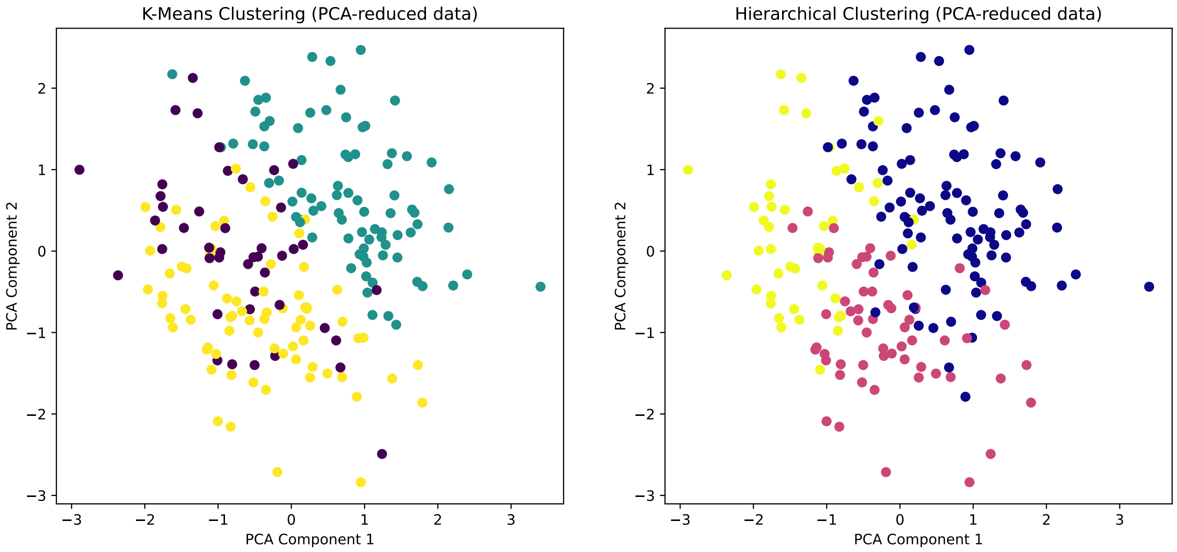
8. Compare K-Means with Hierarchical Clustering

```python
# Plot both K-Means and Hierarchical clusters on the PCA-reduced data
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# K-Means plot
ax1.scatter(pca_data[:, 0], pca_data[:, 1], c=kmeans_labels, cmap='viridis')
ax1.set_title('K-Means Clustering (PCA-reduced data)')
ax1.set_xlabel('PCA Component 1')
ax1.set_ylabel('PCA Component 2')

# Hierarchical clustering plot
ax2.scatter(pca_data[:, 0], pca_data[:, 1], c=agg_labels, cmap='plasma')
ax2.set_title('Hierarchical Clustering (PCA-reduced data)')
ax2.set_xlabel('PCA Component 1')
ax2.set_ylabel('PCA Component 2')

plt.show()
```



9. Conclusion:

Below are the insights:

- The pairplot and correlation heatmap revealed the relationships between variables like exercise time, healthy meals, sleep hours, stress level, and BMI.
- Variables, such as stress level and BMI, showed weaker relationships with other variables, suggesting potential independence in certain wellness attributes.
- K-Means successfully segmented the patients into distinct groups, as demonstrated by the silhouette score (a measure of how well clusters are formed). A higher silhouette score (closer to 1) indicates well-separated and cohesive clusters.
- Agglomerative (hierarchical) clustering also segmented patients into clusters, the silhouette score may reveal that K-Means performed better in terms of distinct segmentation. Hierarchical clustering may still offer useful insights in cases of non-linear relationships.
- PCA effectively reduced the dataset into two principal components, capturing most of the variance (explained variance ratio). This allowed for a visual representation of the clusters in two dimensions.
- The scatter plot of PCA components clearly visualized how well the clusters formed. The clusters from both K-Means and hierarchical methods were distinguishable, though K-Means appeared to have more distinct boundaries.
- The silhouette score comparison highlighted that K-Means clustering had slightly better performance in separating the patient groups, while hierarchical clustering was also able to group patients but with potentially more overlapping clusters.
- K-Means, being more efficient with larger datasets, might be preferred for segmenting patients based on wellness data, whereas hierarchical clustering can offer more granular insights, especially for smaller datasets or in exploratory analysis.
- Based on the clustering, health interventions can be tailored to different patient groups. For instance, patients in clusters with lower exercise time and high BMI could benefit from targeted fitness programs, while those with high stress levels but good physical health might need stress-reduction initiatives.
- Additional analysis with more advanced clustering algorithms (e.g., DBSCAN) or incorporation of more features (such as mental health scores) might reveal deeper insights into patient behavior.

Both clustering methods revealed distinct patient profiles in terms of wellness, but K-Means combined with PCA provided clearer and more actionable groupings.