Untitled27

April 20, 2025

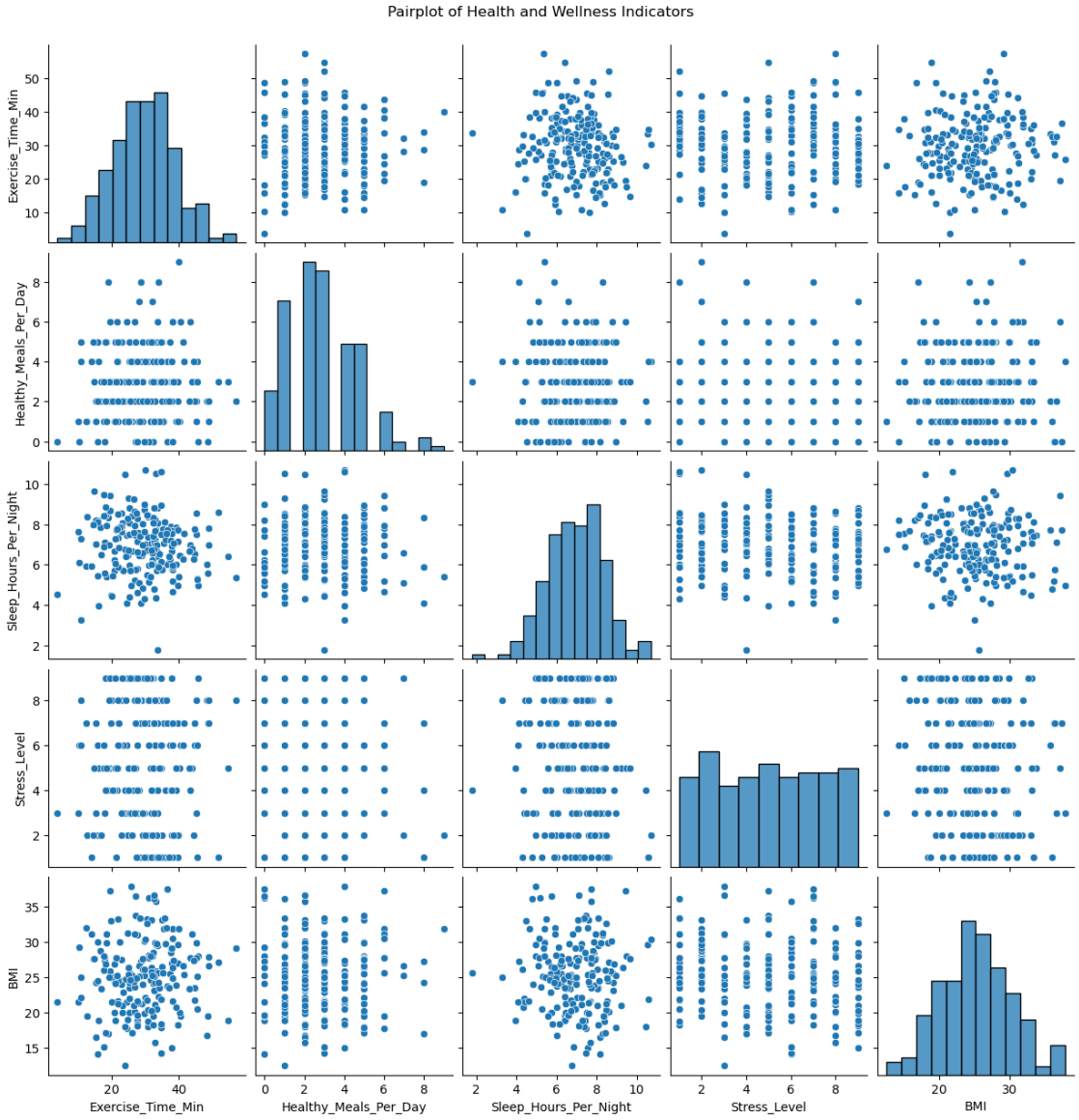
[2]:
```
#  Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score

#  Load the dataset
df = pd.read_csv("simulated_health_wellness_data.csv")

#  EDA: Summary Statistics
print("Summary Statistics:\n", df.describe())

#  EDA: Pairplot
sns.pairplot(df)
plt.suptitle("Pairplot of Health and Wellness Indicators", y=1.02)
plt.show()

#  EDA: Correlation Heatmap
plt.figure(figsize=(8, 6))
correlation = df.corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap of Health and Wellness Indicators")
plt.show()

#  Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df)

#  K-Means Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_labels = kmeans.fit_predict(X_scaled)
kmeans_silhouette = silhouette_score(X_scaled, kmeans_labels)
```

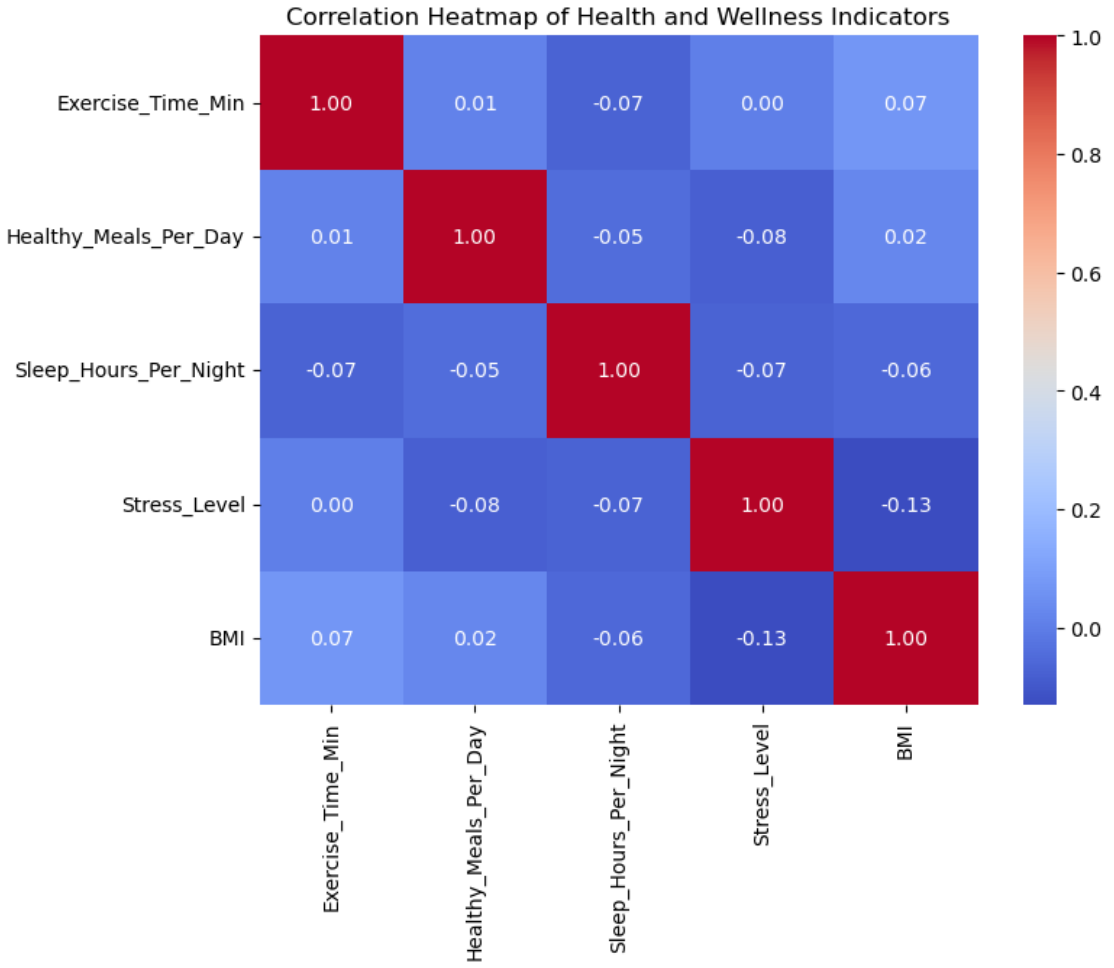
```
#  Agglomerative Clustering
agglo = AgglomerativeClustering(n_clusters=3)
agglo_labels = agglo.fit_predict(X_scaled)
agglo_silhouette = silhouette_score(X_scaled, agglo_labels)

#  PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
pca_explained_variance = pca.explained_variance_ratio_

#  K-Means Clustering After PCA
kmeans_pca = KMeans(n_clusters=3, random_state=42)
kmeans_pca_labels = kmeans_pca.fit_predict(X_pca)
kmeans_pca_silhouette = silhouette_score(X_pca, kmeans_pca_labels)

#  Visualize Clusters Before vs After PCA
plt.figure(figsize=(14, 5))

# Before PCA (K-Means on original data, PCA used for plotting only)
plt.subplot(1, 2, 1)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans_labels, cmap='viridis')
plt.title("K-Means Clustering (Original Data, PCA Visualized)")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")

# After PCA (K-Means on PCA-reduced data)
plt.subplot(1, 2, 2)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans_pca_labels, cmap='plasma')
plt.title("K-Means Clustering (After PCA)")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")

plt.tight_layout()
plt.show()

#  Compare Results
comparison_df = pd.DataFrame({
    "Model": ["K-Means", "Agglomerative", "K-Means after PCA"],
    "Silhouette Score": [kmeans_silhouette, agglo_silhouette,␣
 ↪kmeans_pca_silhouette],
    "Explained Variance by PCA (total)": [np.nan, np.nan,␣
 ↪pca_explained_variance.sum()]
})

print("\nModel Comparison:\n", comparison_df)
```

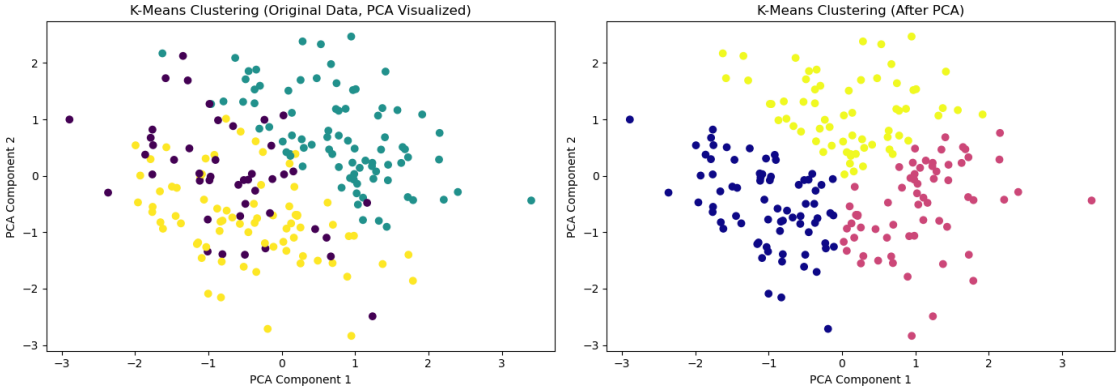
Summary Statistics:

	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


Pairplot of Health and Wellness Indicators
Exercise_Time_Min
Healthy_Meals_Per_Day
Sleep_Hours_Per_Night
Stress_Level
BMI





Model Comparison:

Model	Silhouette Score	Explained Variance by PCA (total)

```
   0            K-Means          0.151616                          NaN
   1      Agglomerative          0.136285                          NaN
   2  K-Means after PCA          0.362561                     0.457741
```

[ ]:

[ ]: