

Project-1

November 15, 2025

```
[1]: import pandas as pd

[2]: from IPython.display import Image, display

[3]: import numpy as np

[84]: import matplotlib.pyplot as pyplot

[10]: import seaborn as sbn

[11]: sbn.set(style="whitegrid")

[19]: df=pd.read_csv("fitness_dataset.csv")

[20]: df.head()
```

	age	height_cm	weight_kg	heart_rate	blood_pressure	sleep_hours	\
0	56	152	65	69.6	117.0	NaN	
1	69	186	95	60.8	114.8	7.5	
2	46	192	103	61.4	116.4	NaN	
3	32	189	83	60.2	130.1	7.0	
4	60	175	99	58.1	115.8	8.0	

	nutrition_quality	activity_index	smokes	gender	is_fit
0	2.37	3.97	no	F	1
1	8.77	3.19	0	F	1
2	8.20	2.03	0	F	0
3	6.18	3.68	0	M	1
4	9.95	4.83	yes	F	1

```
[21]: print("Section 2 - Explore Data")

Section 2 - Explore Data

[22]: print("Dataset shape:", df.shape)

Dataset shape: (2000, 11)

[23]: print("\nData info:", df.info())
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   age                   2000 non-null   int64
1   height_cm             2000 non-null   int64
2   weight_kg             2000 non-null   int64
3   heart_rate            2000 non-null   float64
4   blood_pressure        2000 non-null   float64
5   sleep_hours           1840 non-null   float64
6   nutrition_quality     2000 non-null   float64
7   activity_index        2000 non-null   float64
8   smokes                2000 non-null   object
9   gender                2000 non-null   object
10  is_fit                2000 non-null   int64
dtypes: float64(5), int64(4), object(2)
memory usage: 172.0+ KB

```

Data info: None

```
[24]: print(df.isnull().sum())
```

```

age                0
height_cm          0
weight_kg          0
heart_rate         0
blood_pressure     0
sleep_hours       160
nutrition_quality  0
activity_index     0
smokes             0
gender             0
is_fit            0
dtype: int64

```

```
[25]: print(" Fix Missing Values")
```

Fix Missing Values

```
[26]: missing_h = df.isnull().mean() * 100
```

```
[27]: print(missing_h)
```

```

age                0.0
height_cm          0.0
weight_kg          0.0
heart_rate         0.0
blood_pressure     0.0

```

```

sleep_hours      8.0
nutrition_quality 0.0
activity_index   0.0
smokes           0.0
gender           0.0
is_fit           0.0
dtype: float64

```

```
[29]: cols_to_drop = missing_h[missing_h > 30].index
```

```
[31]: df = df.drop(columns=cols_to_drop)
```

```
[32]: numeric_cols = df.select_dtypes(include=["float64", "int64"]).columns
```

```
[33]: for col in numeric_cols:
        df[col] = df[col].fillna(df[col].median())
```

```
[34]: print(df.isnull().sum())
```

```

age              0
height_cm        0
weight_kg         0
heart_rate        0
blood_pressure    0
sleep_hours       0
nutrition_quality 0
activity_index    0
smokes            0
gender            0
is_fit            0
dtype: int64

```

```
[35]: print("Detect Outliers")
```

```
Detect Outliers
```

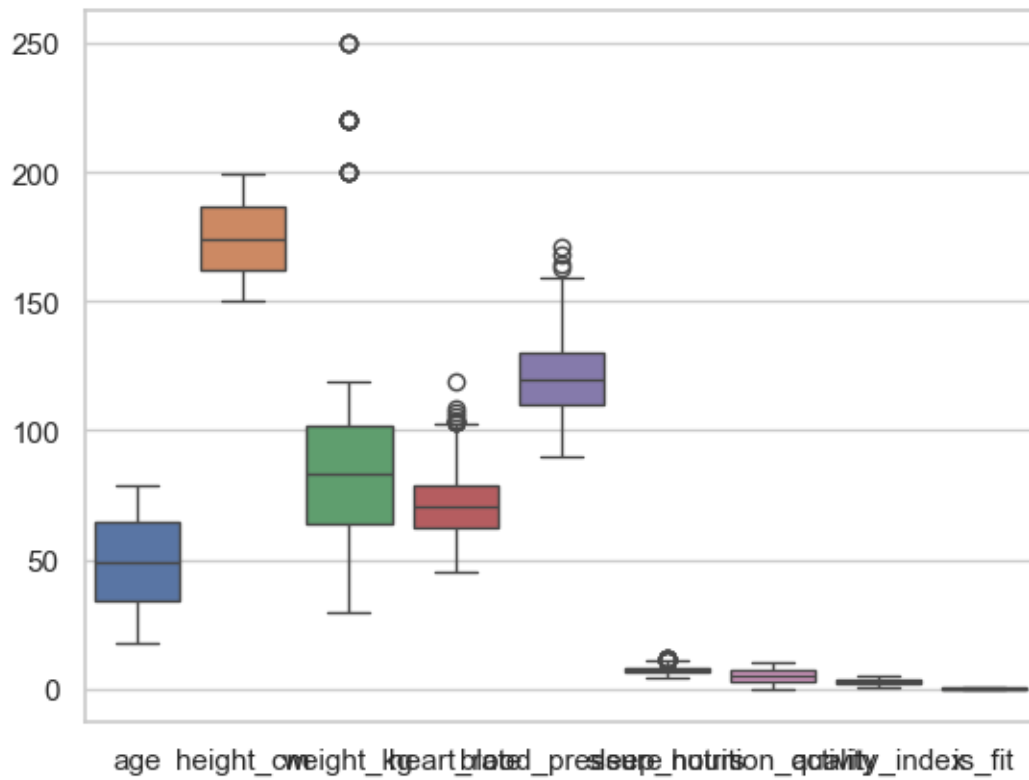
```
[37]: pyplot.figure(figsize=(12,5))
```

```
[37]: <Figure size 1200x500 with 0 Axes>
```

```
<Figure size 1200x500 with 0 Axes>
```

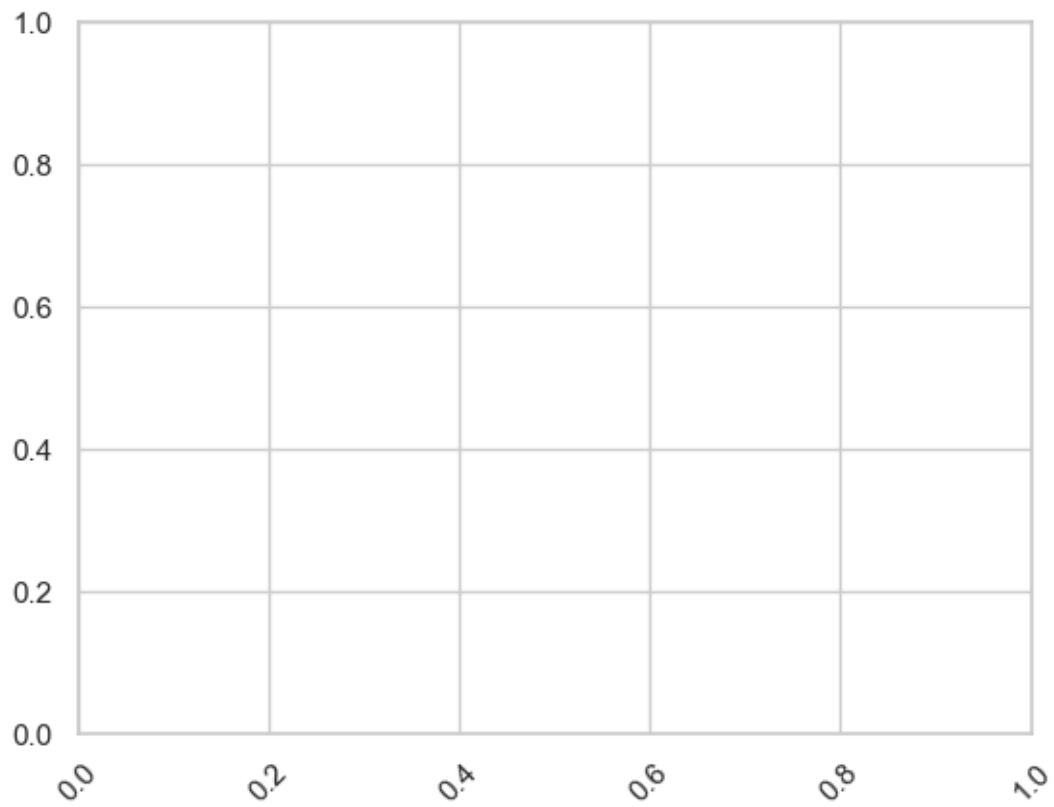
```
[38]: sbn.boxplot(data=df[numeric_cols])
```

```
[38]: <Axes: >
```



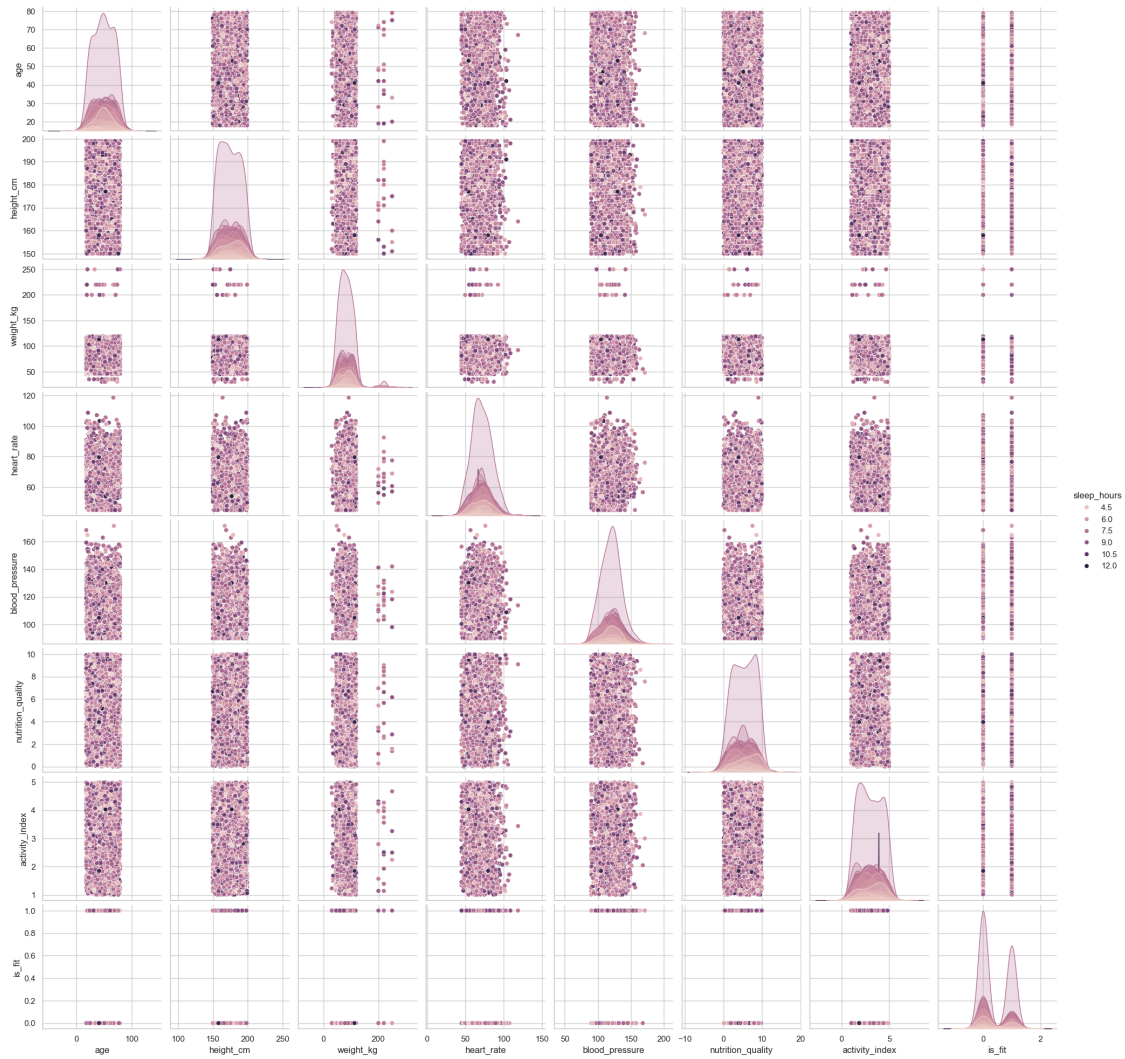
```
[39]: plt.xticks(rotation=45)
```

```
[39]: (array([0. , 0.2, 0.4, 0.6, 0.8, 1. ]),
      [Text(0.0, 0, '0.0'),
       Text(0.2, 0, '0.2'),
       Text(0.4, 0, '0.4'),
       Text(0.6000000000000001, 0, '0.6'),
       Text(0.8, 0, '0.8'),
       Text(1.0, 0, '1.0')])
```



```
[41]: sbn.pairplot(data=df, hue='sleep_hours')
```

```
[41]: <seaborn.axisgrid.PairGrid at 0x117f55950>
```



```
[42]: plt.show()
```

```
[43]: def outliers_iqr(column):
    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    low = Q1 - 1.5 * IQR
    high = Q3 + 1.5 * IQR
    return (column < low) | (column > high)
```

```
[44]: data_iqr = df.copy()
```

```
[45]: for col in numeric_cols:
    data_iqr[col + "_iqr_outlier"] = outliers_iqr(data_iqr[col])
```

```
[46]: data_iqr[data_iqr.filter(like="_iqr_outlier").any(axis=1)].head()
```

```
[46]:
```

	age	height_cm	weight_kg	heart_rate	blood_pressure	sleep_hours	\
72	37	155	93	107.1	110.4	8.0	
102	53	169	78	105.6	108.9	7.4	
132	42	191	48	103.3	108.8	11.5	
263	49	182	200	71.9	122.0	5.9	
268	79	160	250	77.5	141.9	7.7	

	nutrition_quality	activity_index	smokes	gender	is_fit	age_iqr_outlier	\
72	2.70	1.72	yes	M	0	False	
102	0.18	1.51	yes	M	0	False	
132	3.53	2.61	yes	M	0	False	
263	5.43	3.95	0	F	0	False	
268	1.38	4.67	yes	M	0	False	

	height_cm_iqr_outlier	weight_kg_iqr_outlier	heart_rate_iqr_outlier	\
72	False	False	True	
102	False	False	True	
132	False	False	True	
263	False	True	False	
268	False	True	False	

	blood_pressure_iqr_outlier	sleep_hours_iqr_outlier	\
72	False	False	
102	False	False	
132	False	True	
263	False	False	
268	False	False	

	nutrition_quality_iqr_outlier	activity_index_iqr_outlier	\
72	False	False	
102	False	False	
132	False	False	
263	False	False	
268	False	False	

	is_fit_iqr_outlier
72	False
102	False
132	False
263	False
268	False

```
[47]: def outliers_z(col, threshold=3):
      mean = col.mean()
      std = col.std()
```

```

z = (col - mean) / std
return abs(z) > threshold

```

```
[48]: data_z = df.copy()
```

```
[49]: for col in numeric_cols:
      data_z[col + "_z_outlier"] = outliers_z(data_z[col])
```

```
[50]: data_z[data_z.filter(like="_z_outlier").any(axis=1)].head()
```

```
[50]:
```

	age	height_cm	weight_kg	heart_rate	blood_pressure	sleep_hours	\
72	37	155	93	107.1	110.4	8.0	
263	49	182	200	71.9	122.0	5.9	
268	79	160	250	77.5	141.9	7.7	
400	42	182	220	63.4	103.7	8.4	
415	21	198	86	108.7	118.2	8.9	

	nutrition_quality	activity_index	smokes	gender	is_fit	age_z_outlier	\
72	2.70	1.72	yes	M	0	False	
263	5.43	3.95	0	F	0	False	
268	1.38	4.67	yes	M	0	False	
400	5.66	1.93	no	F	0	False	
415	3.12	2.40	0	F	1	False	

	height_cm_z_outlier	weight_kg_z_outlier	heart_rate_z_outlier	\
72	False	False	True	
263	False	True	False	
268	False	True	False	
400	False	True	False	
415	False	False	True	

	blood_pressure_z_outlier	sleep_hours_z_outlier	\
72	False	False	
263	False	False	
268	False	False	
400	False	False	
415	False	False	

	nutrition_quality_z_outlier	activity_index_z_outlier	is_fit_z_outlier
72	False	False	False
263	False	False	False
268	False	False	False
400	False	False	False
415	False	False	False

```
[51]: print("Section 5 - Fix Outliers")
```

Section 5 - Fix Outliers

```
[52]: def replace_outliers(column):
      Q1 = column.quantile(0.25)
      Q3 = column.quantile(0.75)
      IQR = Q3 - Q1
      low = Q1 - 1.5 * IQR
      high = Q3 + 1.5 * IQR
      return column.where((column >= low) & (column <= high), np.nan)
```

```
[53]: clean_data = df.copy()
```

```
[54]: for col in numeric_cols:
      clean_data[col] = replace_outliers(clean_data[col])
```

```
[55]: print(clean_data.isnull().sum())
```

```
age                0
height_cm          0
weight_kg          21
heart_rate         10
blood_pressure      4
sleep_hours        13
nutrition_quality   0
activity_index      0
smokes             0
gender             0
is_fit             0
dtype: int64
```

```
[56]: from sklearn.impute import KNNImputer
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[56], line 1
----> 1 from sklearn.impute import KNNImputer

ModuleNotFoundError: No module named 'sklearn'
```

```
[57]: pip install sklearn
```

```
Collecting sklearn
  Downloading sklearn-0.0.post12.tar.gz (2.6 kB)
  Preparing metadata (setup.py) ... error
error: subprocess-exited-with-error

× python setup.py egg_info did not run successfully.
  exit code: 1
  > [15 lines of output]
```

The 'sklearn' PyPI package is deprecated, use 'scikit-learn' rather than 'sklearn' for pip commands.

Here is how to fix this error in the main use cases:

- use 'pip install scikit-learn' rather than 'pip install sklearn'
- replace 'sklearn' by 'scikit-learn' in your pip requirements files
(requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
- if the 'sklearn' package is used by one of your dependencies, it would be great if you take some time to track which package uses
'sklearn' instead of 'scikit-learn' and report it to their issue tracker
- as a last resort, set the environment variable
SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

More information is available at
<https://github.com/scikit-learn/sklearn-pypi-package>
[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

error: metadata-generation-failed

× Encountered error while generating package metadata.

> See above for output.

note: This is an issue with the package mentioned above, not pip.

hint: See above for details.

Note: you may need to restart the kernel to use updated packages.

```
[58]: from sklearn.impute import KNNImputer
```

```
-----  
ModuleNotFoundError                                Traceback (most recent call last)  
Cell In[58], line 1  
----> 1 from sklearn.impute import KNNImputer  
  
ModuleNotFoundError: No module named 'sklearn'
```

```
[59]: pip install scikit-learn
```

Collecting scikit-learn

Downloading scikit_learn-1.7.2-cp313-cp313-macosx_10_13_x86_64.whl.metadata
(11 kB)

```

Requirement already satisfied: numpy>=1.22.0 in
/opt/anaconda3/envs/CodeLetq/lib/python3.13/site-packages (from scikit-learn)
(2.3.4)
Collecting scipy>=1.8.0 (from scikit-learn)
  Downloading scipy-1.16.3-cp313-cp313-macosx_14_0_x86_64.whl.metadata (62 kB)
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.5.2-py3-none-any.whl.metadata (5.6 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Downloading threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.7.2-cp313-cp313-macosx_10_13_x86_64.whl (9.2 MB)
      9.2/9.2 MB
1.8 MB/s  0:00:05 eta 0:00:010m
Downloading joblib-1.5.2-py3-none-any.whl (308 kB)
Downloading scipy-1.16.3-cp313-cp313-macosx_14_0_x86_64.whl (23.5 MB)
      23.5/23.5 MB
3.3 MB/s  0:00:07m0:00:0100:01
Downloading threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
      4/4 [scikit-learn] [0m [scikit-learn]
Successfully installed joblib-1.5.2 scikit-learn-1.7.2 scipy-1.16.3
threadpoolctl-3.6.0
Note: you may need to restart the kernel to use updated packages.

```

```
[60]: from sklearn.impute import KNNImputer
```

```
[61]: imputer = KNNImputer(n_neighbors=3)
```

```
[62]: data_imputed = pd.DataFrame(
    imputer.fit_transform(clean_data[numeric_cols]),
    columns=numeric_cols
)
```

```
[63]: data_imputed.head()
```

```
[63]:
```

	age	height_cm	weight_kg	heart_rate	blood_pressure	sleep_hours	\
0	56.0	152.0	65.0	69.6	117.0	7.5	
1	69.0	186.0	95.0	60.8	114.8	7.5	
2	46.0	192.0	103.0	61.4	116.4	7.5	
3	32.0	189.0	83.0	60.2	130.1	7.0	
4	60.0	175.0	99.0	58.1	115.8	8.0	

	nutrition_quality	activity_index	is_fit
0	2.37	3.97	1.0
1	8.77	3.19	1.0
2	8.20	2.03	0.0
3	6.18	3.68	1.0
4	9.95	4.83	1.0

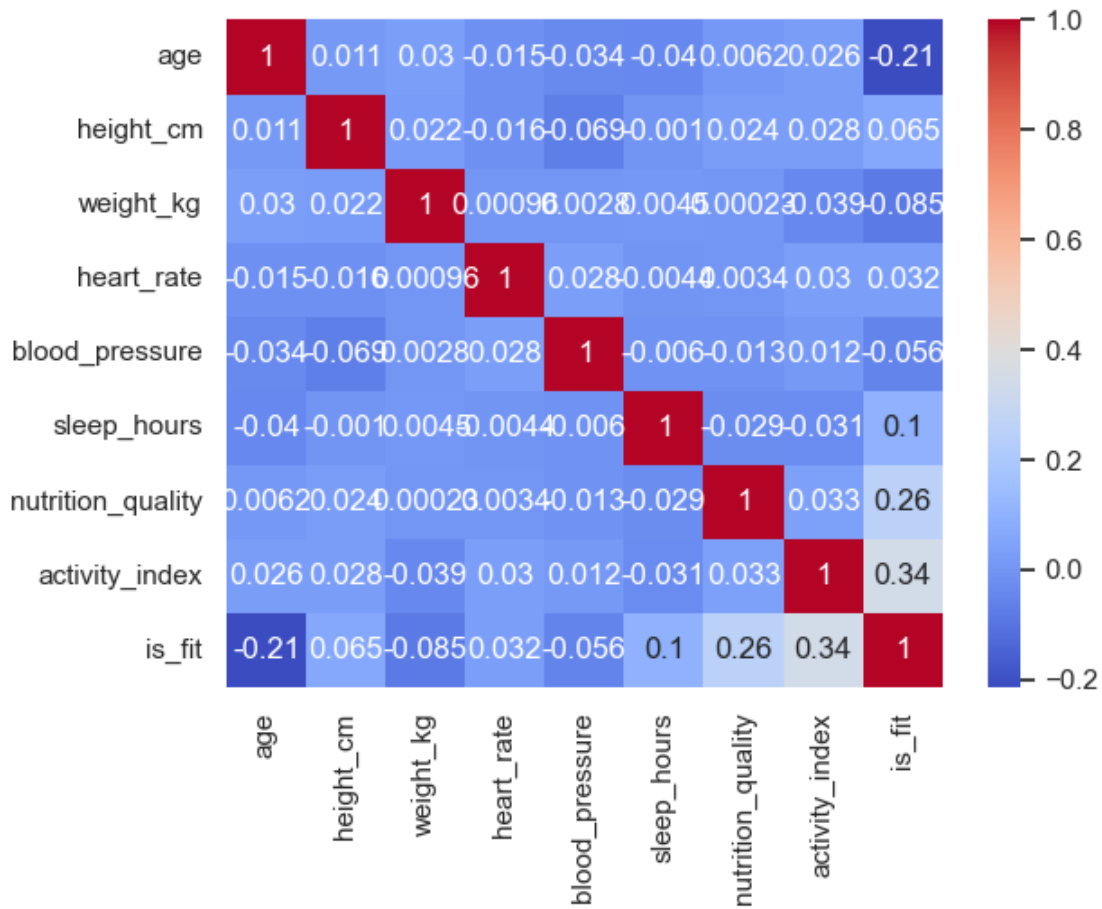
```
[65]: plt.figure(figsize=(10,7))
```

```
[65]: <Figure size 1000x700 with 0 Axes>
```

```
<Figure size 1000x700 with 0 Axes>
```

```
[66]: sns.heatmap(data_imputed.corr(), annot=True, cmap="coolwarm")
```

```
[66]: <Axes: >
```



```
[67]: plt.show()
```

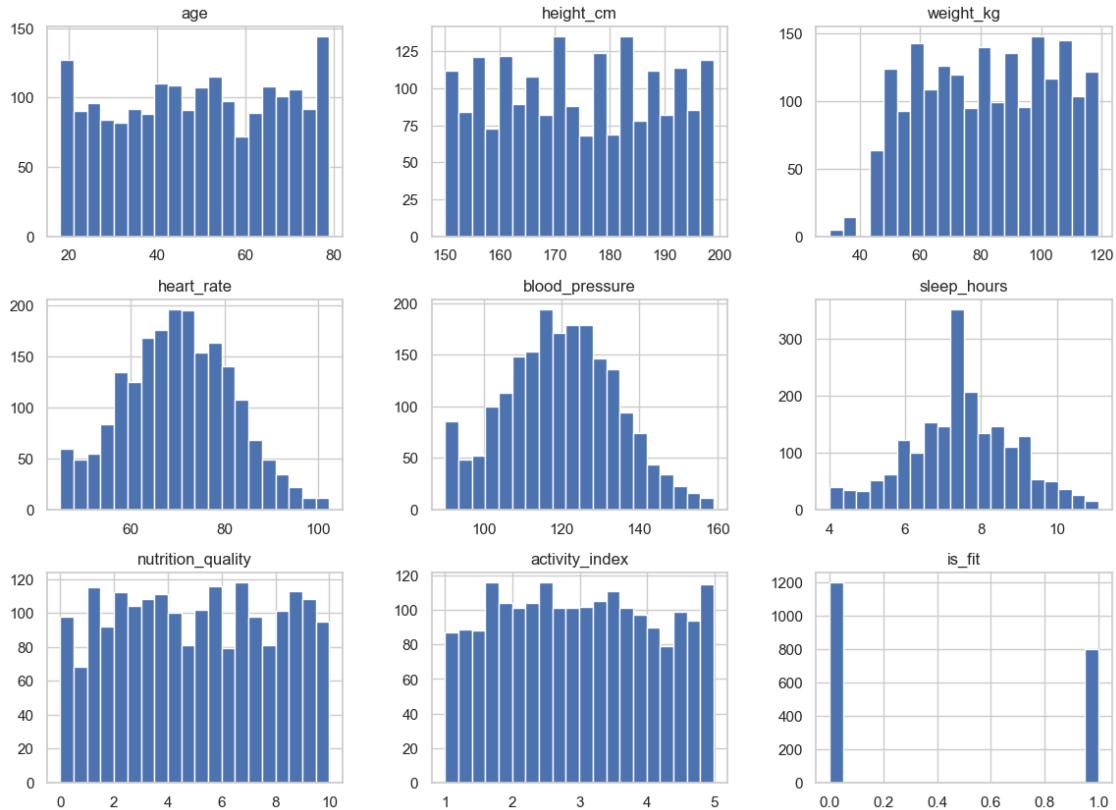
```
[68]: data_imputed.hist(figsize=(14,10), bins=20)
```

```
[68]: array([[<Axes: title={'center': 'age'}>,
<Axes: title={'center': 'height_cm'}>,
<Axes: title={'center': 'weight_kg'}>],
[<Axes: title={'center': 'heart_rate'}>,
<Axes: title={'center': 'blood_pressure'}>],
[<Axes: title={'center': 'sleep_hours'}>,
<Axes: title={'center': 'nutrition_quality'}>],
[<Axes: title={'center': 'activity_index'}>,
<Axes: title={'center': 'is_fit'}>]])
```

```

<Axes: title={'center': 'sleep_hours'}>],
[<Axes: title={'center': 'nutrition_quality'}>,
<Axes: title={'center': 'activity_index'}>,
<Axes: title={'center': 'is_fit'}>]], dtype=object)

```



```
[69]: plt.tight_layout()
```

<Figure size 640x480 with 0 Axes>

```
[70]: plt.show()
```

```
[71]: clean_numeric = data_imputed.copy()
```

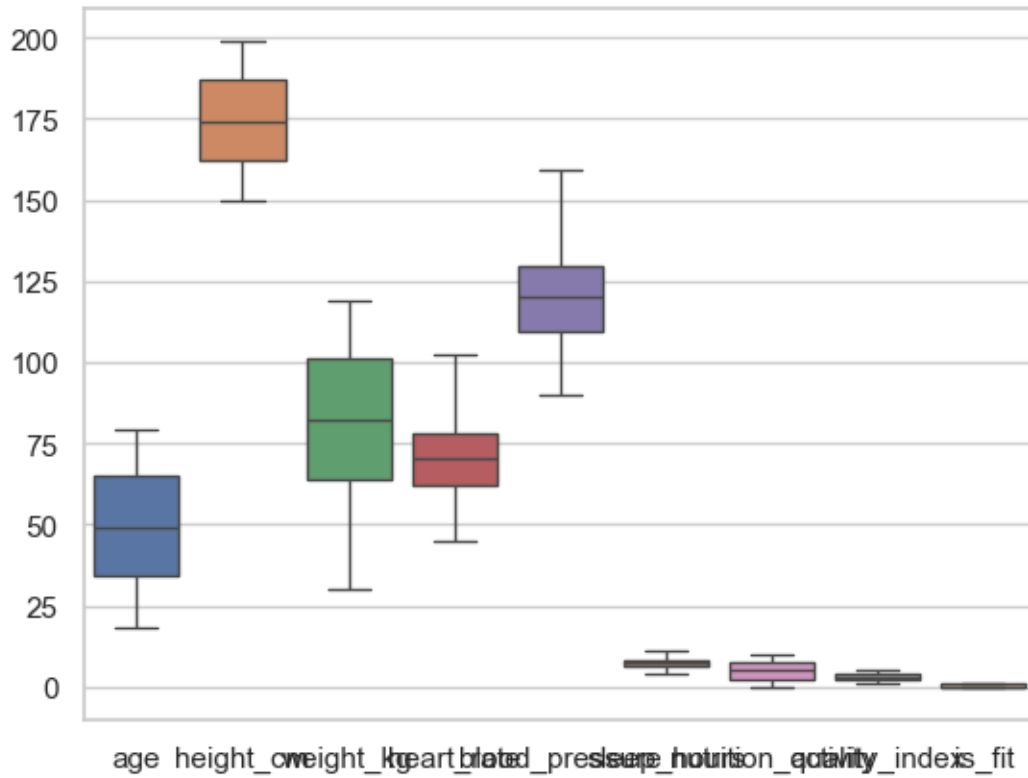
```
[72]: plt.figure(figsize=(12, 5))
```

[72]: <Figure size 1200x500 with 0 Axes>

<Figure size 1200x500 with 0 Axes>

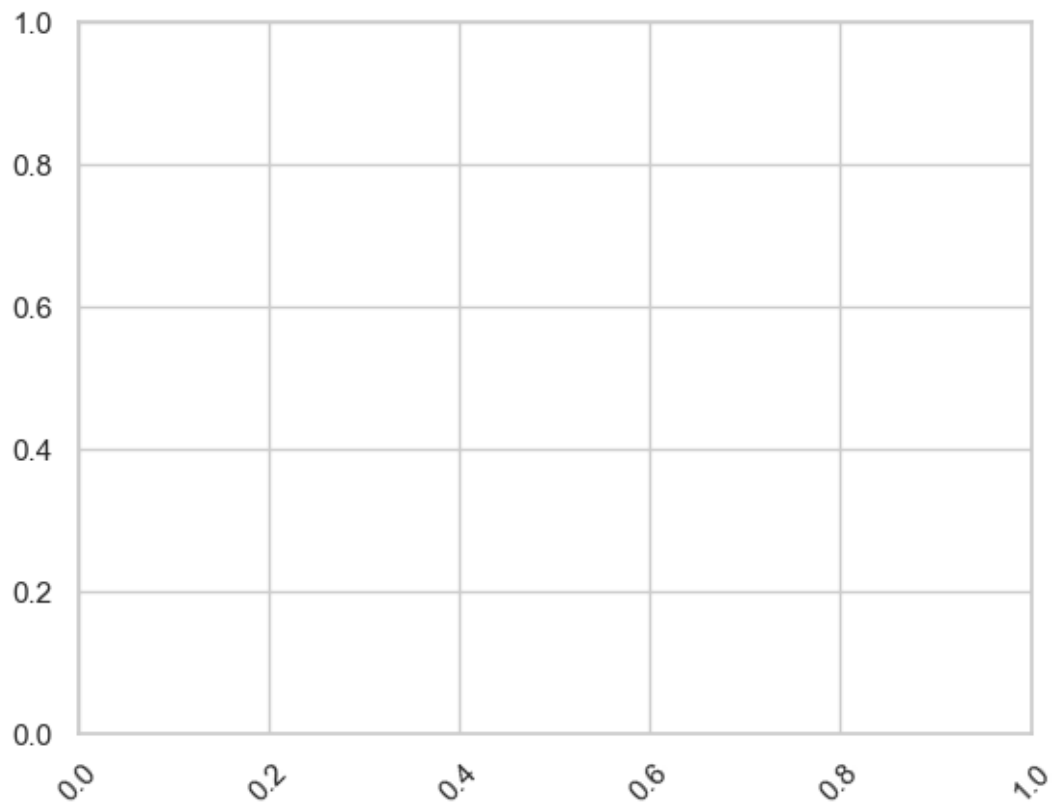
```
[73]: sns.boxplot(data=clean_numeric)
```

[73]: <Axes: >



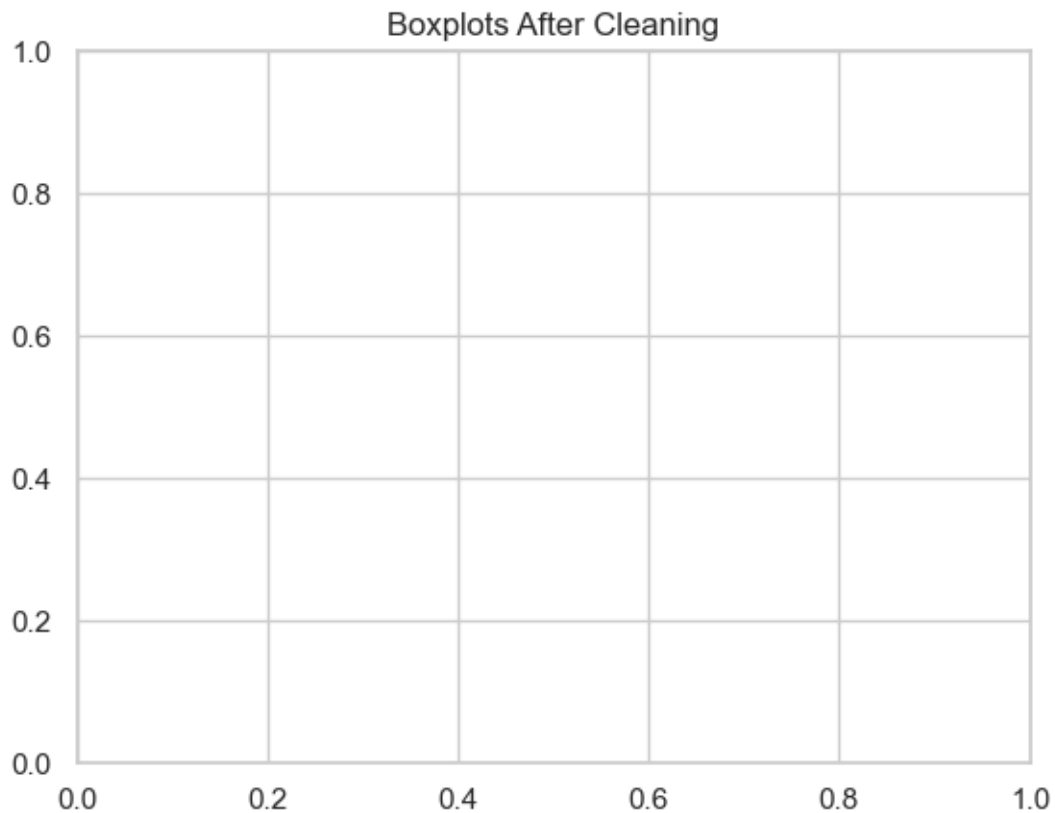
```
[74]: plt.xticks(rotation=45)
```

```
[74]: (array([0. , 0.2, 0.4, 0.6, 0.8, 1. ]),
      [Text(0.0, 0, '0.0'),
       Text(0.2, 0, '0.2'),
       Text(0.4, 0, '0.4'),
       Text(0.6000000000000001, 0, '0.6'),
       Text(0.8, 0, '0.8'),
       Text(1.0, 0, '1.0')])
```



```
[75]: pyplot.title("Boxplots After Cleaning")
```

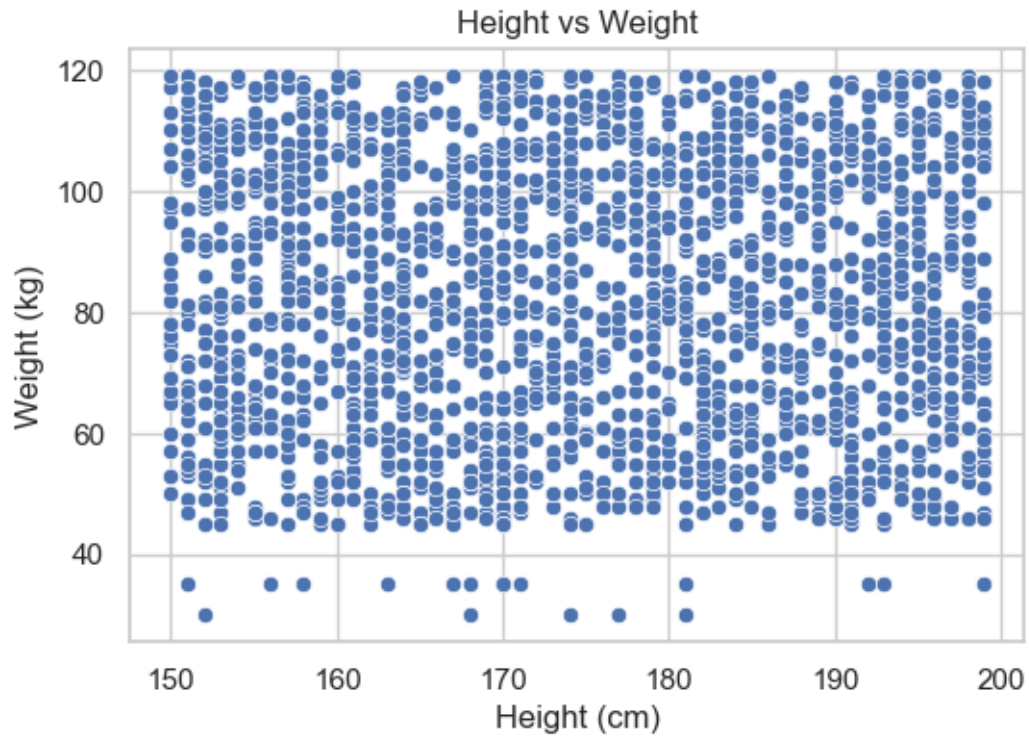
```
[75]: Text(0.5, 1.0, 'Boxplots After Cleaning')
```



```
[76]: plt.show()
```

```
[78]: clean_numeric = data_imputed.copy()

plt.figure(figsize=(6, 4))
sns.scatterplot(x=clean_numeric["height_cm"], y=clean_numeric["weight_kg"])
plt.title("Height vs Weight")
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
plt.show()
```



```
[82]: df["age"].value_counts().sort_index().plot(
      kind="bar", figsize=(6, 4), color="skyblue"
    )
    pyplot.title("Class Distribution")
    pyplot.xlabel("Class")
    pyplot.ylabel("Count")
    pyplot.show()
```

