In [1]:

```python
#23/01/2025

import pandas as pd
data = {
    'Age':[25,None,30,35,None,4.5],
    'Height':[5.5,6.1,5.9,None,5.8,5.2],
    'HelmetStatus':[1,None,1,0,1,1],
    'Gender':['M','F','M',None,'F','M'],
    'Experience':[2,4,6,1,None,None]
}
df=pd.DataFrame(data)

print("Original DataFrame with Missing Values:")
print(df)

#replacing missing values for numerical column
df_filled_mean = df.copy()
df_filled_median = df.copy()
df_filled_mode = df.copy()

#replacing missing values with a mean for numerical colums
df_filled_mean['Age'].fillna(df['Age'].mean(),inplace=True)
df_filled_mean['Height'].fillna(df['Height'].mean(),inplace=True)
df_filled_mean['Experience'].fillna(df['Experience'].mean(),inplace=True)

#replacing missing values with a median for numerical colums
df_filled_median['Age'].fillna(df['Age'].median(),inplace=True)
df_filled_median['Height'].fillna(df['Height'].median(),inplace=True)
df_filled_median['Experience'].fillna(df['Experience'].median(),inplace=True)

#replacing missing values with a mode for numerical colums
df_filled_median['Gender'].fillna(df['Gender'].mode(),inplace=True)
df_filled_median['HelmetStatus'].fillna(df['HelmetStatus'].mode(),inplace=True)

#show the results for all the imputations
print("\nDataFrames After Replacing Missing Values With Mean:")
print(df_filled_mean)

print("\nDataFrames After Replacing Missing Values With Median:")
print(df_filled_median)

print("\nDataFrames After Replacing Missing Values With Mode:")
print(df_filled_mode)
```

```
Original DataFrame with Missing Values:
    Age  Height  HelmetStatus Gender  Experience
0  25.0     5.5           1.0      M         2.0
1   NaN     6.1           NaN      F         4.0
2  30.0     5.9           1.0      M         6.0
3  35.0     NaN           0.0   None         1.0
4   NaN     5.8           1.0      F         NaN
5   4.5     5.2           1.0      M         NaN

DataFrames After Replacing Missing Values With Mean:
      Age  Height  HelmetStatus Gender  Experience
0  25.000     5.5           1.0      M        2.00
1  23.625     6.1           NaN      F        4.00
2  30.000     5.9           1.0      M        6.00
3  35.000     5.7           0.0   None        1.00
4  23.625     5.8           1.0      F        3.25
5   4.500     5.2           1.0      M        3.25

DataFrames After Replacing Missing Values With Median:
    Age  Height  HelmetStatus Gender  Experience
0  25.0     5.5           1.0      M         2.0
1  27.5     6.1           NaN      F         4.0
2  30.0     5.9           1.0      M         6.0
3  35.0     5.8           0.0    NaN         1.0
4  27.5     5.8           1.0      F         3.0
5   4.5     5.2           1.0      M         3.0

DataFrames After Replacing Missing Values With Mode:
    Age  Height  HelmetStatus Gender  Experience
0  25.0     5.5           1.0      M         2.0
1   NaN     6.1           NaN      F         4.0
2  30.0     5.9           1.0      M         6.0
3  35.0     NaN           0.0   None         1.0
4   NaN     5.8           1.0      F         NaN
5   4.5     5.2           1.0      M         NaN
```

```
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:22: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_mean['Age'].fillna(df['Age'].mean(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:23: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_mean['Height'].fillna(df['Height'].mean(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:24: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_mean['Experience'].fillna(df['Experience'].mean(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:27: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_median['Age'].fillna(df['Age'].median(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:28: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_median['Height'].fillna(df['Height'].median(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:29: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_median['Experience'].fillna(df['Experience'].median(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:32: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_median['Gender'].fillna(df['Gender'].mode(),inplace=True)
C:\Users\gundr\AppData\Local\Temp\ipykernel_18540\3948977065.py:33: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_filled_median['HelmetStatus'].fillna(df['HelmetStatus'].mode(),inplace=True)
```

In [ ]: