```
# ========================================
# HR Analytics Dataset - Data Cleaning & Metrics
# ========================================
import pandas as pd
from sklearn.preprocessing import LabelEncoder
# Step 1: Load raw CSV file
# Make sure the CSV file is in the same folder or give full path
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
# Step 2: Standardize column names
df.columns = (
    df.columns.str.strip()            # remove leading/trailing spaces
              .str.replace(" ", "_")   # replace spaces with underscores
              .str.lower()             # convert to lowercase
)
# Step 3: Basic data check
print("✅ Dataset Shape:", df.shape)
print("\n🔍 Missing Values:\n", df.isnull().sum())
# Step 4: Handle missing values
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = df[col].fillna(df[col].mode()[0])
    else:
        df[col] = df[col].fillna(df[col].median())
# Step 5: Separate numeric & categorical columns
numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns.tolist()
categorical_cols = df.select_dtypes(include=['object']).columns.tolist()
# Step 6: Convert categorical to category type
for col in categorical_cols:
    df[col] = df[col].astype('category')


# Step 7: Encode categorical variables for analysis
```

```
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Step 8: Create new metrics
# Attrition column after encoding will be numeric now (Yes=1, No=0)
df['attrition_flag'] = df['attrition']

# Example: tenure_years before current role
df['tenure_years'] = df['totalworkingyears'] - df['yearsatcompany']

# Step 9: Summary metrics
attrition_rate = df['attrition_flag'].mean() * 100
avg_tenure = df['yearsatcompany'].mean()
avg_income = df['monthlyincome'].mean()

print(f"\n📊 Attrition Rate: {attrition_rate:.2f}%")
print(f"📊 Average Tenure: {avg_tenure:.2f} years")
print(f"📊 Average Monthly Income: ${avg_income:,.2f}")

# Step 10: Save cleaned dataset
df.to_csv("cleaned_hr_dataset.csv", index=False)
print("\n✅ Cleaned data saved as 'cleaned_hr_dataset.csv'")
```