```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

mh_df = pd.read_csv('/content/Mental_Health_Dataset/mental_health_dataset.csv')
mh_df.head(5)
```

_		age	gender	employment_status	work_environment	mental_health_history	seeks_treatment	stress_level	sleep_hours	physical_activit
	0	56	Male	Employed	On-site	Yes	Yes	6	6.2	
	1	46	Female	Student	On-site	No	Yes	10	9.0	
	2	32	Female	Employed	On-site	Yes	No	7	7.7	
	3	60	Non- binary	Self-employed	On-site	No	No	4	4.5	
	4	25	Female	Self-employed	On-site	Yes	Yes	3	5.4	

```
# Inspecting the dataset
print(mh_df.shape)
mh_df.info()
→ (10000, 14)
     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 10000 entries, 0 to 9999
     Data columns (total 14 columns):
     # Column
                                Non-Null Count Dtype
                                10000 non-null int64
     0
         age
                                10000 non-null object
         gender
     1
         employment_status
                                10000 non-null object
         work_environment
                                10000 non-null object
         mental_health_history 10000 non-null object
         seeks_treatment
                                10000 non-null object
         stress_level
                                10000 non-null int64
                                10000 non-null float64
         sleep hours
         physical_activity_days 10000 non-null int64
         depression_score
                                10000 non-null int64
     10 anxiety score
                                10000 non-null
                                               int64
     11 social_support_score
                                10000 non-null int64
                                10000 non-null float64
     12 productivity_score
     13 mental_health_risk
                                10000 non-null object
     dtypes: float64(2), int64(6), object(6)
```

Observations of the Data Inspection

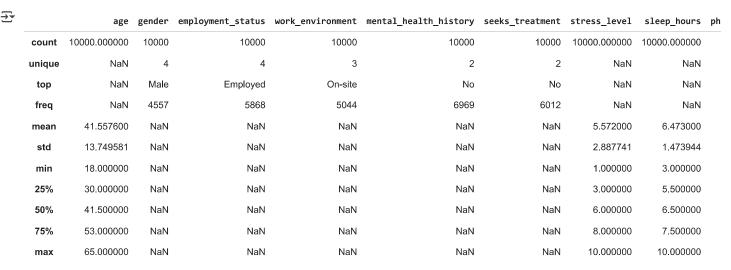
memory usage: 1.1+ MB

- The dataset has a total of 10,000 row entries with 14 different variables
- · No missing values were observed
- · The data types for all columns were in their correct format.

```
# Check for duplicate rows
mh_df.duplicated().sum()

→ np.int64(0)

# Summary statistics for all variables
# mh_df.describe(include='all')
```



Observations of the Summary Statistics

Check for Outliers:

8

9

Medium

Medium

- The mean value of all *continuous variables* (age, stress_level, sleep_hours, physical_activity_days, depression_score, anxiety_score, social_support_score, and productivity_score) in the dataset was approximately equal to the median, hence indicating a symmetric/ normally distributed data.
- No apparent outliers were observed based on the minimum, maximum, interquartile range, and standard deviation. The values appear to fall within a reasonable range, suggesting consistency across the dataset.
- Summary statistics across the dataset reveal that most variables in the dataset demonstrate high variability (CV > 50%), as evidenced by their standard deviation values relative to their respective means, while a few exhibit moderate to low variation.

```
# Visualise Outliers using Box plot for all number variables
# Selecting the numerical columns
selected_cols = ['age', 'stress_level', 'sleep_hours', 'physical_activity_days', 'depression_score', 'anxiety_score', 'social_support_score'
melted_df = mh_df[selected_cols].melt(var_name='Variable', value_name='Value')
# Plotting it
plt.figure(figsize=(15, 8))
sns.boxplot(x='Variable', y='Value', data=melted_df)
plt.title("Boxplots for Multiple Variables")
plt.savefig('boxplot_for_multiple_variables.png')
plt.show()
      Show hidden output
# Encoding the ordinal values in the "mental_health_risk" variable with numbers for future spearman correlation analysis
# The mapping defined
risk_mapping = {'Low': 1, 'Medium': 2, 'High': 3}
mh_df['mental_health_risk_encoded'] = mh_df['mental_health_risk'].map(risk_mapping)
mh_df.to_csv('encoded_dataset.csv', index=False)
print(mh_df[['mental_health_risk', 'mental_health_risk_encoded']].head(10))
       mental_health_risk mental_health_risk_encoded
₹
                     High
                     High
                                                     3
     2
                   Medium
                                                     2
                                                     1
     3
                      Low
     4
                     High
                                                     3
                   Medium
     6
                   Medium
                                                     2
     7
                   Medium
                                                     2
```

2

2

Start coding or generate with AI.

Start coding or $\underline{\text{generate}}$ with AI.