Practical **5**

```
# Sub : Essential Technologies for Data Science
# Name: Kunal Tushar Mahale
# Msc Data Science
# Practical 5
```

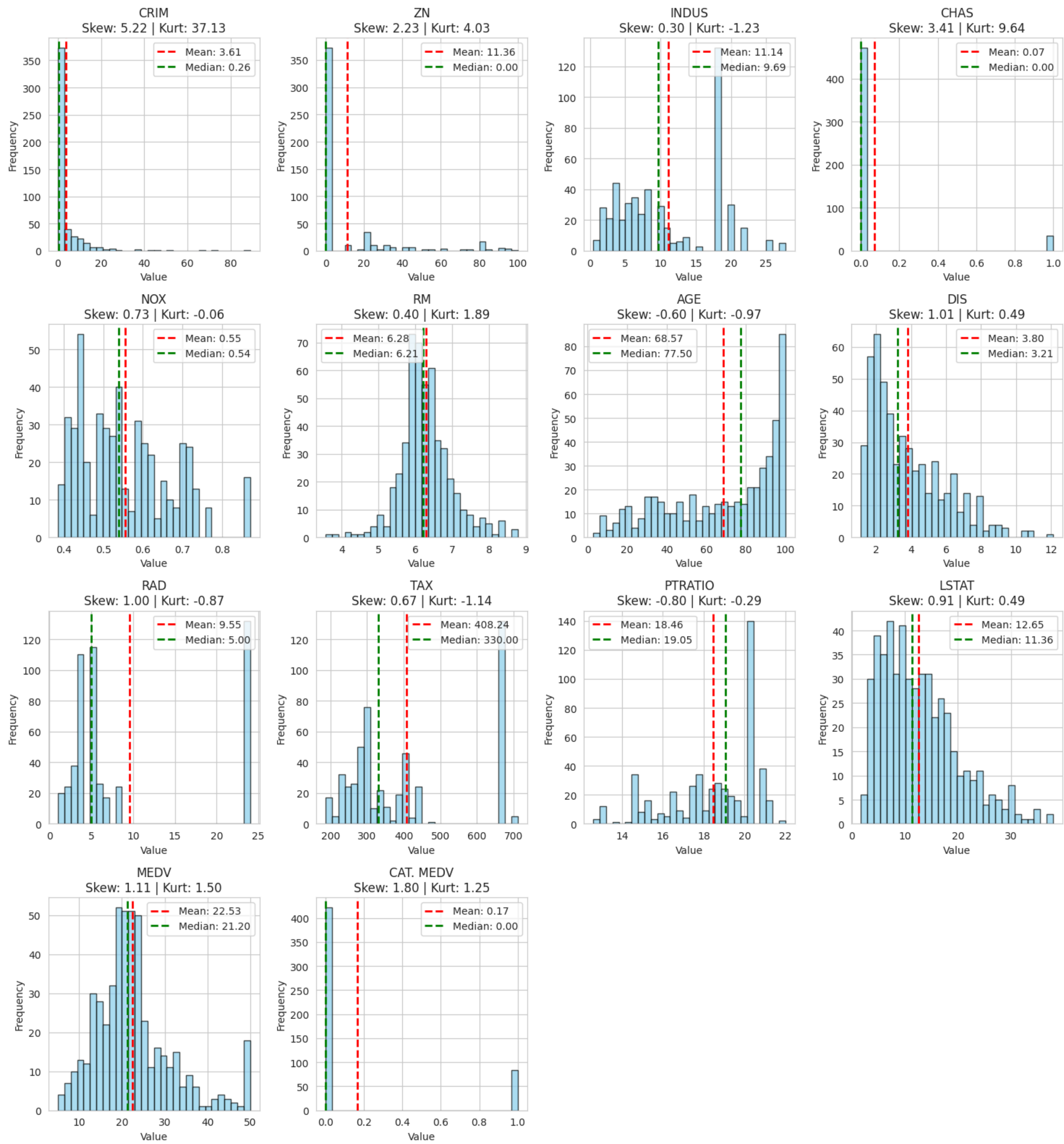
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#1. Load Boston Csv file in python
csv = pd.read_csv('/content/BostonHousing.csv')
```

```
# 2. univariate analysis on the numeric columns
print("Univariate Analysis")
print(csv[numeric_cols].describe())
```

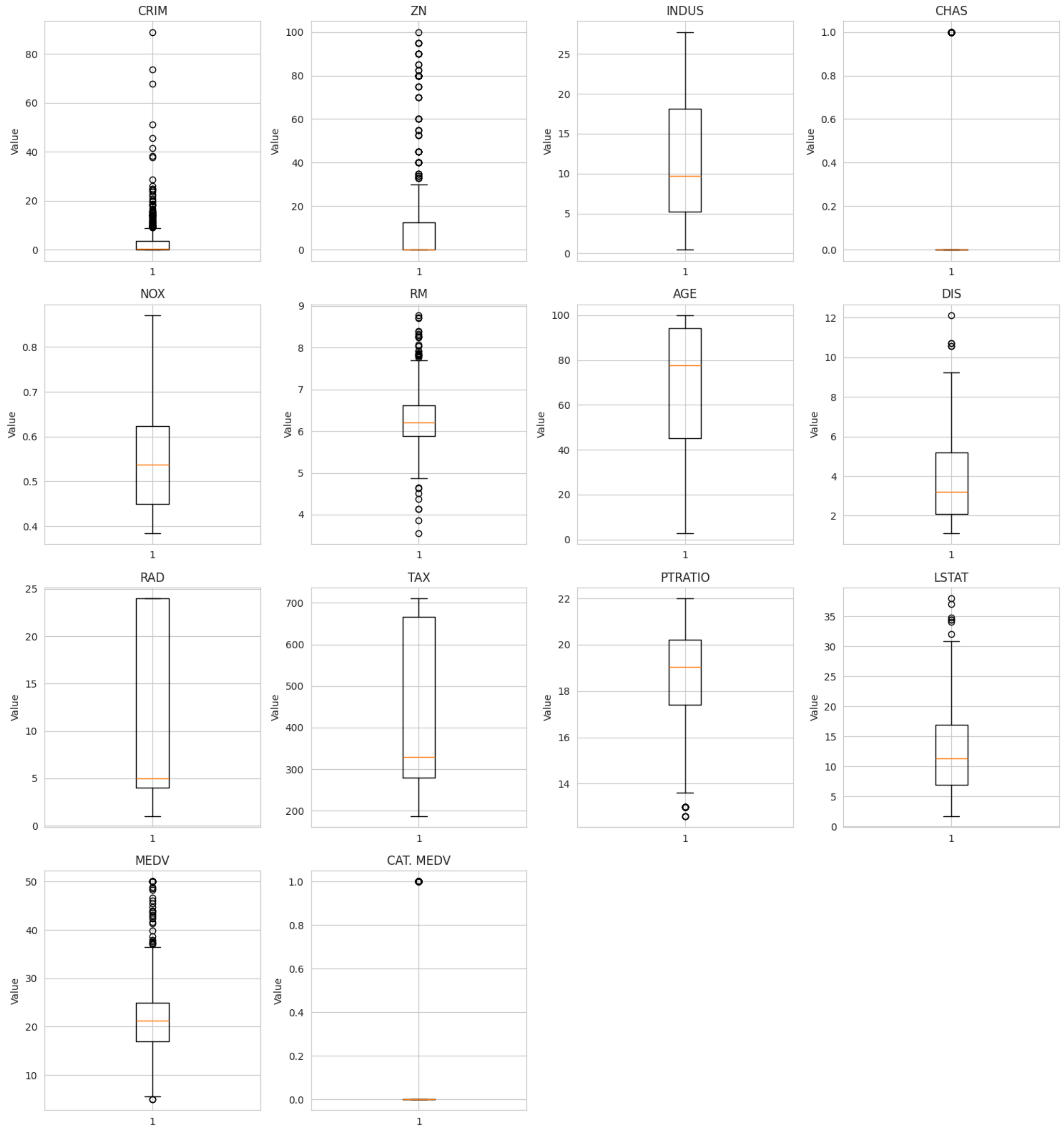
```
Univariate Analysis
             CRIM          ZN       INDUS        CHAS         NOX          RM  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695    6.284634
std      8.601545   23.322453    6.860353    0.253994    0.115878    0.702617
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%      0.082045    0.000000    5.190000    0.000000    0.449000    5.885500
50%      0.256510    0.000000    9.690000    0.000000    0.538000    6.208500
75%      3.677083   12.500000   18.100000    0.000000    0.624000    6.623500
max     88.976200  100.000000   27.740000    1.000000    0.871000    8.780000

              AGE         DIS         RAD         TAX     PTRATIO       LSTAT  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean    68.574901    3.795043    9.549407  408.237154   18.455534   12.653063
std     28.148861    2.105710    8.707259  168.537116    2.164946    7.141062
min      2.900000    1.129600    1.000000  187.000000   12.600000    1.730000
25%     45.025000    2.100175    4.000000  279.000000   17.400000    6.950000
50%     77.500000    3.207450    5.000000  330.000000   19.050000   11.360000
75%     94.075000    5.188425   24.000000  666.000000   20.200000   16.955000
max    100.000000   12.126500   24.000000  711.000000   22.000000   37.970000

             MEDV   CAT. MEDV
count  506.000000  506.000000
mean    22.532806    0.166008
std      9.197104    0.372456
min      5.000000    0.000000
25%     17.025000    0.000000
50%     21.200000    0.000000
75%     25.000000    0.000000
max     50.000000    1.000000
```

```
# 3. HISTOGRAM PLOTS WITH KDE
n_cols = len(numeric_cols)
n_rows = (n_cols + 2) // 4  # 4 plots per row

fig, axes = plt.subplots(n_rows, 4, figsize=(15, n_rows * 4))
axes = axes.flatten()

for i, col in enumerate(numeric_cols):
    # Plot histogram
    axes[i].hist(csv[col], bins=30, alpha=0.7, color='skyblue', edgecolor='black')

    # Add title with stats
    mean_val = csv[col].mean()
    median_val = csv[col].median()
    skew_val = csv[col].skew()
    kurt_val = csv[col].kurtosis()

    axes[i].set_title(f'{col}\nSkew: {skew_val:.2f} | Kurt: {kurt_val:.2f}')

    axes[i].set_xlabel('Value')

    axes[i].set_ylabel('Frequency')

    # Add mean and median lines
    axes[i].axvline(mean_val, color='red', linestyle='--', linewidth=2, label=f'Mean: {mean_val:.2f}')
    axes[i].axvline(median_val, color='green', linestyle='--', linewidth=2, label=f'Median: {median_val:.2f}')
    axes[i].legend()
# Hide extra subplots
for i in range(n_cols, len(axes)):
    axes[i].axis('off')

plt.tight_layout()
plt.savefig('histogram_plots.png')
plt.show()
```




```
# 4. BOX PLOTS
fig, axes = plt.subplots(n_rows, 4, figsize=(15, n_rows * 4))
axes = axes.flatten()

for i, col in enumerate(numeric_cols):
    # Plot boxplot
    axes[i].boxplot(csv[col].dropna())
    axes[i].set_title(f'{col}')
    axes[i].set_ylabel('Value')

# Hide extra subplots
for i in range(n_cols, len(axes)):
    axes[i].axis('off')

plt.tight_layout()
```

```
plt.savefig('boxplots.png')
plt.show()
```




```
# 5. INFERENCES
print("INFERENCES FOR EACH COLUMN")

for col in numeric_cols:

    skew = csv[col].skew()

    kurt = csv[col].kurtosis()
```

```
    print(f"{col}:")

    # Skewness interpretation

    if skew > 1:
        print(f"  Skewness: {skew:.3f} - Highly Right Skewed (Long right tail)")
    elif skew > 0.5:
        print(f"  Skewness: {skew:.3f} - Moderately Right Skewed")
    elif skew > -0.5:
        print(f"  Skewness: {skew:.3f} - Approximately Symmetric (Normal)")
    elif skew > -1:
        print(f"  Skewness: {skew:.3f} - Moderately Left Skewed")
    else:
        print(f"  Skewness: {skew:.3f} - Highly Left Skewed (Long left tail)")

    # Kurtosis interpretation

    if kurt > 3:
        print(f"  Kurtosis: {kurt:.3f} - Heavy Tails (Many outliers)")
    elif kurt > -1:
        print(f"  Kurtosis: {kurt:.3f} - Normal Tails")
    else:
        print(f"  Kurtosis: {kurt:.3f} - Light Tails (Few outliers)")

    # Count outliers using IQR method

    Q1 = csv[col].quantile(0.25)
    Q3 = csv[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = csv[(csv[col] < lower_bound) | (csv[col] > upper_bound)]

    print(f"  Outliers: {len(outliers)} ({len(outliers)/len(csv)*100:.2f}%)")
    print()
```

```
  Skewness: 2.226 - Highly Right Skewed (Long right tail)
  Kurtosis: 4.032 - Heavy Tails (Many outliers)
  Outliers: 68 (13.44%)

INDUS:
  Skewness: 0.295 - Approximately Symmetric (Normal)
  Kurtosis: -1.234 - Light Tails (Few outliers)
  Outliers: 0 (0.00%)

CHAS:
  Skewness: 3.406 - Highly Right Skewed (Long right tail)
  Kurtosis: 9.638 - Heavy Tails (Many outliers)
  Outliers: 35 (6.92%)

NOX:
  Skewness: 0.729 - Moderately Right Skewed
  Kurtosis: -0.065 - Normal Tails
  Outliers: 0 (0.00%)

RM:
  Skewness: 0.404 - Approximately Symmetric (Normal)
  Kurtosis: 1.892 - Normal Tails
  Outliers: 30 (5.93%)

AGE:
  Skewness: -0.599 - Moderately Left Skewed
  Kurtosis: -0.968 - Normal Tails
  Outliers: 0 (0.00%)

DIS:
  Skewness: 1.012 - Highly Right Skewed (Long right tail)
  Kurtosis: 0.488 - Normal Tails
  Outliers: 5 (0.99%)

RAD:
  Skewness: 1.005 - Highly Right Skewed (Long right tail)
  Kurtosis: -0.867 - Normal Tails
  Outliers: 0 (0.00%)

TAX:
  Skewness: 0.670 - Moderately Right Skewed
  Kurtosis: -1.142 - Light Tails (Few outliers)
  Outliers: 0 (0.00%)

PTRATIO:
  Skewness: -0.802 - Moderately Left Skewed
  Kurtosis: -0.285 - Normal Tails
  Outliers: 15 (2.96%)

LSTAT:
  Skewness: 0.906 - Moderately Right Skewed
  Kurtosis: 0.493 - Normal Tails
  Outliers: 7 (1.38%)

MEDV:
  Skewness: 1.108 - Highly Right Skewed (Long right tail)
  Kurtosis: 1.495 - Normal Tails
  Outliers: 40 (7.91%)
```

```
# 6. SUMMARY
print("OVERALL SUMMARY")
skewed_features = sum(1 for col in numeric_cols if abs(csv[col].skew()) > 1)
heavy_tail_features = sum(1 for col in numeric_cols if csv[col].kurtosis() > 3)

print(f"Total Numeric Features Analyzed: {len(numeric_cols)}")

print(f"Highly Skewed Features: {skewed_features}")
print(f"Heavy Tailed Features: {heavy_tail_features}")
print("Note: Highly skewed features may need transformation before modeling")
```

```
OVERALL SUMMARY
Total Numeric Features Analyzed: 14
Highly Skewed Features: 7
Heavy Tailed Features: 3
Note: Highly skewed features may need transformation before modeling
```