```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import skew, kurtosis
from scipy.stats import bootstrap

def load_data(file_path, skip_rows=4):
    """Load dataset from a CSV file, skipping the first few rows."""
    data = pd.read_csv(file_path, skiprows=skip_rows)
    return data

def clean_data(data, missing_threshold=0.8):
    """Remove columns with many missing values."""
    threshold = len(data) * missing_threshold
    cleaned_data = data.dropna(thresh=threshold, axis=1)
    return cleaned_data

def visualize_distribution(data, column, year, figsize=(8, 6)):
    """Plot distribution of a column for a specific year."""
    plt.figure(figsize=figsize)
    sns.histplot(data[column].dropna(), kde=True)
    plt.title(f'Distribution of {column} in {year}')
    plt.xlabel(f'{column} (% of total population)')
    plt.ylabel('Frequency')
    plt.show()

def visualize_boxplot(data, columns, year, figsize=(12, 8)):
    """Plot boxplot to visualize the distribution of urban population by year."""
    plt.figure(figsize=figsize)
    sns.boxplot(data=data[columns], orient='v')
```

```
    plt.title(f'Distribution of Urban Population by Year in {year}')
    plt.xlabel('Year')
    plt.ylabel('Urban population (% of total population)')
    plt.xticks(rotation=45)
    plt.show()

def visualize_country_trend(data, country, years, figsize=(10, 6)):
    """Plot trend of urban population for a specific country."""
    country_data = data[data['Country Name'] == country].iloc[0, 4:]
    plt.figure(figsize=figsize)
    plt.plot(years, country_data)
    plt.title(f'Urban Population Trend in {country}')
    plt.xlabel('Year')
    plt.ylabel('Urban population (% of total population)')
    plt.xticks(rotation=45)
    plt.grid(True)
    plt.show()

def visualize_correlation_heatmap(data, countries, figsize=(10, 8)):
    """Plot correlation heatmap between years for selected countries."""
    selected_countries_data = data[data['Country Name'].isin(countries)]
    corr_matrix_countries = selected_countries_data.iloc[:, 4:].corr()
    plt.figure(figsize=figsize)
    sns.heatmap(corr_matrix_countries, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title('Correlation Heatmap for Selected Countries')
    plt.show()

def visualize_top_countries_trends(data, n_countries, figsize=(12, 8)):
    """Plot trends of urban population for top N populated countries."""
    top_countries = data.groupby('Country Name').max().nlargest(n_countries, '2022').index
    plt.figure(figsize=figsize)
```

```
    for country in top_countries:
        country_data = data[data['Country Name'] == country]
        plt.plot(years, country_data.iloc[0, 4:], label=country)

    plt.title(f'Urban Population Trends in Top {n_countries} Populated Countries')
    plt.xlabel('Year')
    plt.ylabel('Urban population (% of total population)')
    plt.legend()
    plt.xticks(rotation=45)
    plt.grid(True)
    plt.show()

def bootstrap_analysis(data, column, confidence_level=0.90, nboot=10000):
    """Perform bootstrap analysis on a column of data."""
    values = data[column].dropna().to_numpy()
    low, high = bootstrap(values, np.mean, confidence_level=confidence_level, nboot=nboot)
    sigma = 0.5 * (high - low)
    print(f"Bootstrap Analysis for {column}:")
    print(f"Average = {np.round(np.mean(values), 4)} +/- {np.round(sigma, 4)}")
    print(f"Standard Deviation = {np.round(np.std(values), 4)} +/- {np.round(sigma, 4)}")
    print(f"Skewness = {np.round(skew(values), 4)}")
    print(f"Kurtosis = {np.round(kurtosis(values), 4)}")

# Load the dataset
data = load_data('population.csv')

# Clean the data
cleaned_data = clean_data(data)

# Get the years
years = cleaned_data.columns[4:]
```

```
# Visualize Distribution for a Selected Year
visualize_distribution(cleaned_data, '2010', 'Urban Population')

# Visualize Boxplot for Urban Population by Year
visualize_boxplot(cleaned_data, years, 'Urban Population')

# Visualize Urban Population Trend for a Specific Country
visualize_country_trend(cleaned_data, 'India', years)

# Visualize Correlation Heatmap for Selected Countries
selected_countries = ['China', 'United States', 'India', 'Brazil', 'United Kingdom']
visualize_correlation_heatmap(cleaned_data, selected_countries)

# Visualize Trends for Top 5 Populated Countries
visualize_top_countries_trends(cleaned_data, 5)

# Perform Bootstrap Analysis for Urban Population
bootstrap_analysis(cleaned_data, '2010')
```