**Comprehensive Guide to Statistical Operations in Data Analysis**

This guide provides detailed explanations and Python code examples for essential statistical operations used in analyzing the features.csv dataset. Each section includes a definition, purpose, interpretation, and code snippet.

### 1. Measures of Central Tendency

These describe the center of the data distribution.

#### 1.1 Mean (Average)

* **Definition**: Sum of values divided by count.
* **Purpose**: Represents the average value.
* **Use case**: Understanding the typical fuel price across all weeks.
* **Code**: df['Fuel\_Price'].mean()

#### 1.2 Median

* **Definition**: Middle value when data is sorted.
* **Purpose**: Robust to outliers, gives better central value in skewed data.
* **Use case**: More accurate central value when Fuel\_Price has extreme values.
* **Code**: df['Fuel\_Price'].median()

#### 1.3 Mode

* **Definition**: Most frequent value.
* **Purpose**: Identifies the most common occurrence.
* **Use case**: Identifying the most common store ID.
* **Code**: df['Store'].mode()

### 2. Measures of Dispersion

These measure how spread out the data is.

#### 2.1 Range

* **Definition**: Difference between maximum and minimum.
* **Purpose**: Shows total spread of values.
* **Code**: df['Temperature'].max() - df['Temperature'].min()

#### 2.2 Variance

* **Definition**: Average of squared differences from the mean.
* **Purpose**: Measures variability.
* **Code**: df['Temperature'].var()

#### 2.3 Standard Deviation

* **Definition**: Square root of variance.
* **Purpose**: Indicates average distance from the mean.
* **Code**: df['Temperature'].std()

#### 2.4 Interquartile Range (IQR)

* **Definition**: Difference between Q3 (75th percentile) and Q1 (25th percentile).
* **Purpose**: Captures middle 50% of data, helps identify outliers.
* **Code**:

Q1 = df['Fuel\_Price'].quantile(0.25)  
Q3 = df['Fuel\_Price'].quantile(0.75)  
IQR = Q3 - Q1

### 3. Shape of Distribution

Describes the symmetry and tail behavior of the data.

#### 3.1 Skewness

* **Definition**: Measure of asymmetry in distribution.
* **Purpose**: Detects whether data is skewed left or right.
* **Interpretation**:
  + Positive: Right-skewed
  + Negative: Left-skewed
* **Code**: df['Fuel\_Price'].skew()

#### 3.2 Kurtosis

* **Definition**: Measure of tails’ weight compared to normal distribution.
* **Purpose**: Detects outliers and peak shape.

## **Interpretation**:

* 3: Heavy tails (leptokurtic)
  + <3: Light tails (platykurtic)
* **Code**: df['Fuel\_Price'].kurtosis()

### 4. Relative Dispersion

#### 4.1 Coefficient of Variation (CV)

* **Definition**: Standard deviation divided by the mean, in %.
* **Purpose**: Compares variability relative to magnitude.
* **Code**:

cv = (df['Fuel\_Price'].std() / df['Fuel\_Price'].mean()) \* 100

### 5. Outlier Detection and Standardization

#### 5.1 Z-Score

* **Definition**: Number of standard deviations a value is from the mean.
* **Purpose**: Identifies outliers.
* **Interpretation**: Values with z > 3 or z < -3 are considered outliers.
* **Code**:

from scipy.stats import zscore  
z\_scores = zscore(df['Fuel\_Price'])  
outliers = df[(z\_scores > 3) | (z\_scores < -3)]

### 6. Correlation

#### 6.1 Pearson Correlation

* **Definition**: Measures linear relationship between two variables.
* **Range**: [-1, 1]
* **Purpose**: Identify dependencies between variables.
* **Code**:

correlation\_matrix = df.corr()

### 7. Group-based Statistical Analysis

#### 7.1 Grouped Mean/STD

* **Purpose**: Compare groups like holidays vs non-holidays.
* **Code**:

df.groupby('IsHoliday')['Fuel\_Price'].agg(['mean', 'std'])

#### 7.2 Group by Store

* **Purpose**: See how economic indicators vary by store.
* **Code**:

df.groupby('Store')[['CPI', 'Unemployment']].mean()

### 8. Handling Missing Values (Statistical Approach)

#### 8.1 Fill with Mean or Median

* **Use when**: Data is assumed missing at random.
* **Code**:

df['CPI'].fillna(df['CPI'].mean(), inplace=True)

#### 8.2 Forward Fill (Time-Series)

* **Use when**: Previous value logically continues.
* **Code**:

df['CPI'].fillna(method='ffill', inplace=True)

### 9. Visualization (Statistical Insights)

#### 9.1 Box Plot

* **Purpose**: Shows IQR, median, outliers.
* **Code**:

import seaborn as sns  
sns.boxplot(x='IsHoliday', y='Fuel\_Price', data=df)

#### 9.2 Histogram + KDE

* **Purpose**: Visualize distribution shape.
* **Code**:

sns.histplot(df['Temperature'], kde=True)

#### 9.3 Heatmap (Correlation Matrix)

* **Purpose**: Visual overview of correlations.
* **Code**:

import matplotlib.pyplot as plt  
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')  
plt.show()

This guide offers a detailed breakdown of statistical operations to support robust data analysis, especially useful for structured datasets like features.csv.