Cheat Sheet for Data Analysis #2Descriptive Statistics

Basic Dataset Inspection

Load a CSV file into a DataFrame

```
df = pd.read_csv("Diamond.csv")
```

Meaning: Reads data from a CSV file and stores it in a variable called `df`

View the shape of the DataFrame (rows, columns)

```
print(df.shape)
```

Meaning: `.shape` returns a tuple: (number of rows, number of columns); `df.shape[0]` means rows and `df.shape[1]` means columns

Display data types of each column

```
print(df.dtypes)
```

Meaning: `.dtypes` shows the data type (e.g., object, int64, float64) of each column

Check for missing values

```
print(df.isnull().sum())
```

Meaning: Shows the count of missing values for each column - essential to check before analysis

Get concise summary of the DataFrame

```
print(df.info())
```

Meaning: `.info()` shows column names, non-null counts, data types, and memory usage

Show the first few rows of the DataFrame

```
print(df.head(10))
```

Meaning: `.head(n)` displays the first n rows (default is 5) - useful for quick inspection

---

Descriptive Statistics for Numerical Variables

Sample size (number of observations)

```
n_obs = len(df[['carat', 'price']])
print(f"Sample size: {n_obs}")
```

Meaning: Counts the total number of records in the dataset for numerical analysis

Minimum values

```
print(df[['carat', 'price']].min())
```

Meaning: Shows the smallest value for each numerical variable - helps identify potential outliers

Maximum values

```
print(df[['carat', 'price']].max())
```

Meaning: Shows the largest value for each numerical variable - helps identify potential outliers

Sample mean (average)

```
print(df[['carat', 'price']].mean())
```

Meaning: Calculates the arithmetic average; represents the "center of mass" of the data but can be influenced by outliers

Sample variance

```
print(df[['carat', 'price']].var(ddof=1))
```

Meaning: Measures the average squared deviation from the mean; `ddof=1` specifies sample variance (N-1 degrees of freedom)

Sample standard deviation

```
print(df[['carat', 'price']].std(ddof=1))
```

Meaning: Square root of variance; in same units as original data; higher value means data points are more spread out

Median

```
print(df[['carat', 'price']].median())
```

Meaning: The middle value when data is sorted; robust to outliers; better represents "typical" value in skewed distributions

Quartiles and Quantiles

```
print(df[['carat', 'price']].quantile([0.25, 0.50, 0.75]))
```

Meaning: Returns values at specified percentiles; Q1 (25%), Q2/median (50%), Q3 (75%); IQR = Q3-Q1 measures spread of middle 50% of data

Skewness

```
print(df[['carat', 'price']].skew())
```

Meaning: Measures asymmetry of distribution; >0 indicates right-skewed (long tail to right), <0 indicates left-skewed

Kurtosis

```
print(df[['carat', 'price']].kurtosis())
```

Meaning: Measures "peakedness" and tail heaviness; >0 indicates heavy tails (more outliers), <0 indicates light tails

---

Descriptive Statistics for Categorical Variables

Frequency tables

```
print(df['colour'].value_counts().sort_index())
```

Meaning: `.value_counts()` shows how many observations fall into each category; reveals most/least common categories

Multiple column frequency counts

```
print(df[['colour', 'certification']].value_counts())
```

Meaning: Counts occurrences of unique combinations across multiple categorical columns

Mode (most frequent value)

```
print(df[['colour', 'clarity', 'certification']].mode().iloc[0])
```

Meaning: Identifies the most frequently occurring category; useful for understanding dominant categories in categorical data

---

Bivariate Analysis

Covariance matrix

```
print(df[['carat', 'price']].cov())
```

Meaning: Measures how two numerical variables change together; positive = variables increase together, negative = inverse relationship; magnitude depends on units

Correlation between two variables

```
print(df['carat'].corr(df['price']))
```

Meaning: Measures strength and direction of linear relationship (-1 to 1); close to 1 = strong positive linear relationship; does not imply causation

Correlation matrix (all numerical variables)

```
print(df[['carat', 'price']].corr())
```

Meaning: Shows pairwise correlations between all numerical variables; diagonal is always 1; high off-diagonal values indicate strong linear relationships

Contingency table (two categorical variables)

```
print(pd.crosstab(df['colour'], df['certification']))
```

Meaning: Shows joint frequency distribution of two categorical variables; reveals associations between categories; large differences suggest potential relationship

---

Practical Tips

1. **Always check for missing data** before performing statistical analysis
2. **Compare mean and median** - if mean > median, distribution is likely right-skewed
3. **Use median and IQR** for skewed distributions instead of mean and standard deviation
4. **Visualize your data** with histograms and box plots to complement numerical statistics
5. **Remember**: correlation does not imply causation - it only measures linear relationships