Exploratory Data Analysis (EDA)

CHEAT SHEET

EVERY DATA ANALYSTS MUST KNOW

UNDERSTAND • EXPLORE • ANALYZE DATA

RAMA GOPALA KRISHNA MASANI

What is EDA?

Exploratory Data Analysis (EDA) is the process of:

- Understanding the structure of data
- Identifying patterns & trends
- Detecting missing values & outliers
- Preparing data for modeling & dashboards

👉 EVERY DATA ANALYST MUST MASTER EDA.

Import Required Libraries

```
1   import pandas as pd
2   import numPy as np
```

Pandas is the core library for EDA in Python

Loading Data (All Common Sources)

```
pd.read_csv('file.csv')
pd.read_excel('file.xlsx')
pd.read_sql(query, connection)
pd.read_json('file.json')
pd.read_html(url)
```

First step in any EDA process.

Basic Data Overview

```
1   df.head()
2   df.tail()
3   df.sample()
4   df.shape
5   df.info()
6   df.describe()
```

Understand rows, columns, datatypes & summary statistics.

Understanding Columns & Data Types

```
1   df.columns
2   df.dtypes
3   df.index
```

Helps identify numerical, categorical & datetime columns.

Selecting Data

```
1   df['column']
2   df[['col1','col2']]
3   df.iloc[0]
4   df.loc[0]
5   df.iloc[0,0]
6   df.loc[0,'column']
```

Used for column & row-level analysis

Filtering Data

```
1   df[df['col'] > 5]
2   df.query('col > 5')
3   df.filter(like='sales')
```

Focus only on relevant data.

Handling Missing Values

```
df.isnull().sum()
df.notnull()
df.dropna()
df.fillna(value)
```

Missing data must be handled before analysis.

Data Cleaning Operations

```
df.drop_duplicates()
df.rename(columns={'old':'new'})
df.astype('int')
df.replace(1,'One')
df.reset_index()
```

Clean data = accurate insights.

Sorting & Sampling

```
df.sort_values('col')
df.sort_values('col', ascending=False)
df.sample(5)
df.nlargest(3,'col')
df.nsmallest(3,'col')
```

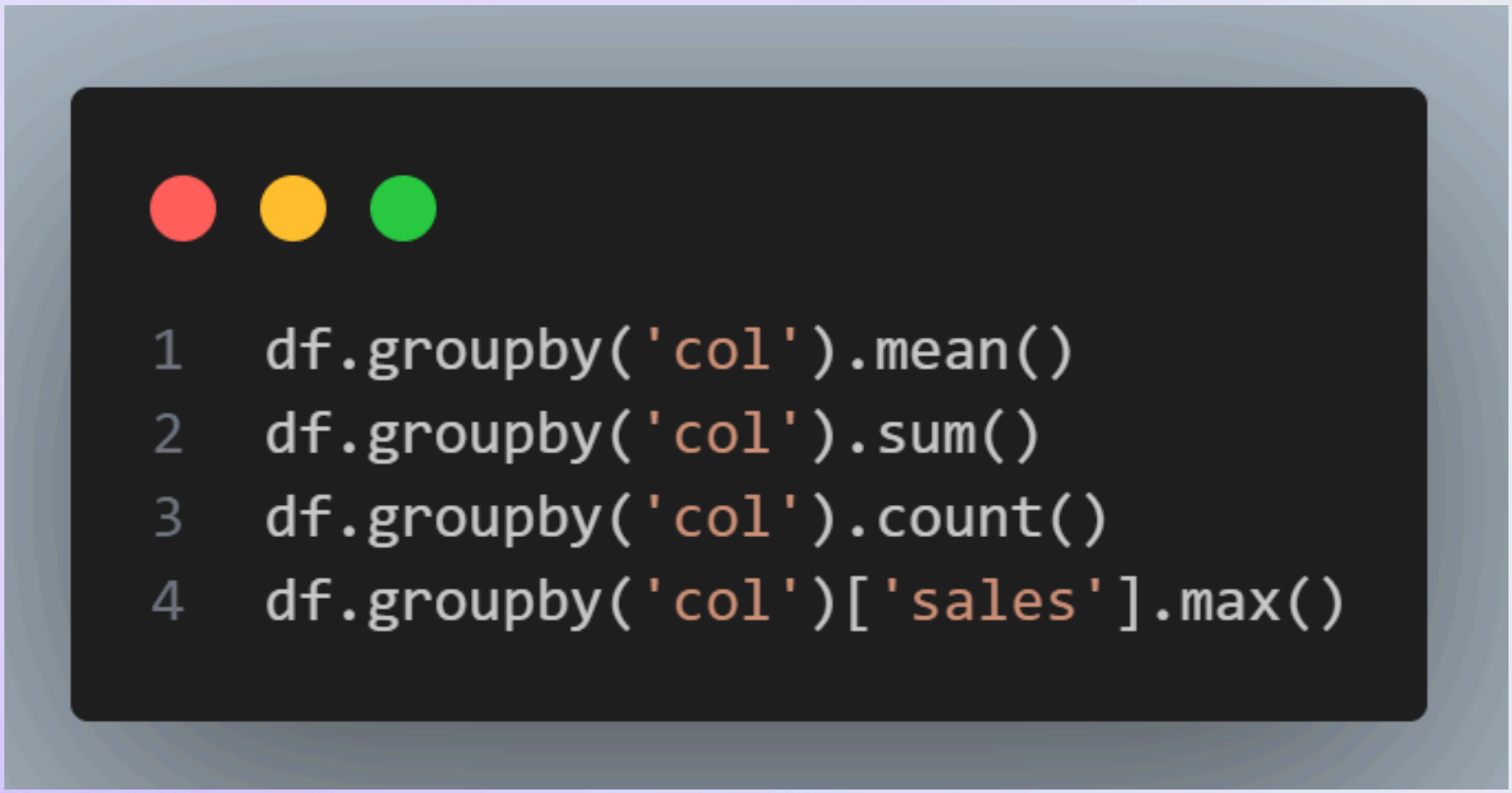
Useful for ranking & trend analysis.

Aggregations & Statistics

```
df.mean()
df.median()
df.std()
df.min()
df.max()
df.sum()
df.corr()
```

Core statistical understanding of data.

GroupBy (Most Important for Analysts)



Used in almost every business analysis.

Pivot Tables

```
pd.pivot_table(
df,
values='sales',
index='region',
aggfunc='mean'
)
```

Excel-like analysis using Python.

Merging & Combining Data

```
pd.concat([df1, df2])
df1.merge(df2, on='key', how='inner')
df1.merge(df2, on='key', how='left')
```

Required for multi-table analysis.

Apply & Transform

```
df.apply(np.mean)
df.transform(lambda x: x + 10)
```

Custom column-wise operations.

Advanced Missing Value Analysis

```
df.isnull().sum()
df.isnull().mean()*100
```

Identify columns with high missing percentage.

Outlier Detection

```
df.describe(percentiles=[.01,.05,.95,.99])
```

Helps detect extreme values before modeling.

Distribution Analysis

```
df['col'].value_counts()
df['col'].nunique()
```

Understand frequency & cardinality.

Date & Time EDA

```
df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
```

Time-based trend analysis.

Categorical Data Analysis

```
df['category'].value_counts(normalize=True)
```

Essential for business insights.

Numerical vs Categorical Analysis

```
df.groupby('category')['sales'].mean()
```

Most common real-world business analysis.

Duplicate Analysis

```
df.duplicated().sum()
```

Prevents inflated metrics.

Feature Engineering

```
df['profit'] = df['revenue'] - df['cost']
```

Convert raw data into meaningful features.

Scaling Awareness

```
df.describe()
```

Helps decide normalization or standardization.

Visualization (Quick Business EDA)

```
df.plot(kind='line')
df.plot(kind='bar')
df.plot(kind='box')
df.plot(kind='hist')
```

Fast insight before dashboards.

Thank You!

Was this helpful?

Follow me for More