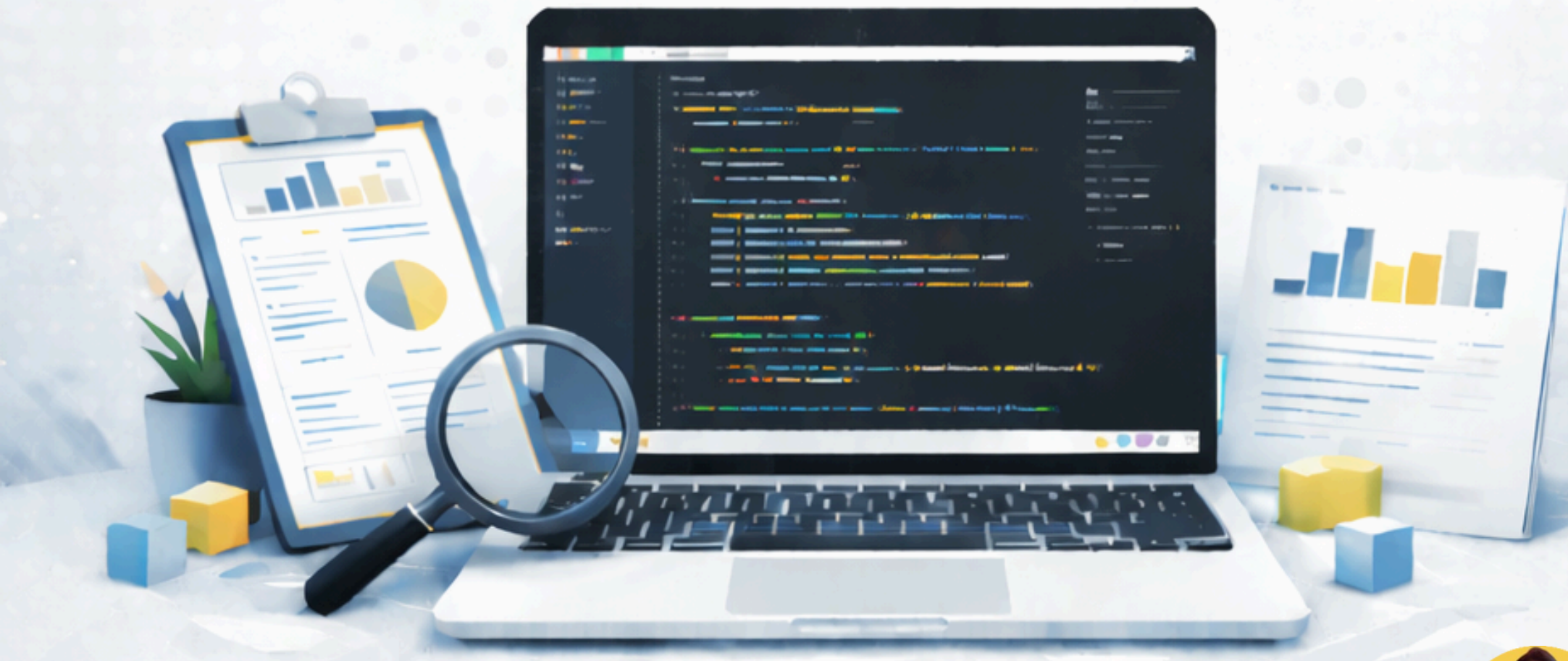
Exploratory Data Analysis (EDA) in Python

Core Techniques Every Data Professional Should Know



Pooja Pawar

EDA Setup

Create virtual env → isolate project
$ python -m venv eda_env

Activate env → start workspace
$ source eda_env/bin/activate

Upgrade pip → avoid install errors
$ python -m pip install --upgrade pip

Install core libraries → EDA essentials
$ pip install pandas numpy
$ pip install matplotlib seaborn

Install stats tools → deeper analysis
$ pip install scipy

Install profiling → automated EDA
$ pip install ydata-profiling

Launch Jupyter → interactive analysis
$ jupyter notebook

Pooja Pawar

Load Data & First Look

Read CSV file → load dataset into a DataFrame
pd.read_csv("data.csv")

Read Excel file → import spreadsheet data into Python
pd.read_excel("data.xlsx")

Head → preview first few rows of the dataset
df.head()

Tail → inspect last rows to check data completeness
df.tail()

Shape → check number of rows and columns
df.shape

Columns → list all feature names in the dataset
df.columns

Pooja Pawar

Data Types & Columns

Dtypes → check column data types
`df.dtypes`

Convert to datetime → fix dates
`pd.to_datetime(df["date"])`

Convert to numeric → clean numbers
`pd.to_numeric(df["amount"], errors="coerce")`

Astype category → optimize memory
`df["city"].astype("category")`

Rename columns → consistency
`df.rename(columns={"Order Date":"order_date"})`

Sort values → inspect extremes
`df.sort_values("amount")`

Unique values → detect IDs
`df.nunique()`

Pooja Pawar

Missing Values Handling

Is null → missing check
df.isna().any()

Null count → column-wise
df.isna().sum()

Null percentage → severity
df.isna().mean()*100

Drop null rows → strict cleaning
df.dropna()

Fill with value → simple impute
df.fillna(0)

Fill with median → numeric fix
df["amount"].fillna(df["amount"].median())

Forward fill → time series
df.fillna(method="ffill")

Pooja Pawar

Duplicates & Quality Checks

Find duplicates → detect repeats
df.duplicated()

Count duplicates → data quality
df.duplicated().sum()

Drop duplicates → clean data
df.drop_duplicates()

Subset duplicates → key-based
df.duplicated(subset=["id","date"])

Memory usage → dataset size
df.memory_usage(deep=True)

Sample rows → random check
df.sample(5)

Value counts → category spread
df["status"].value_counts()

Pooja Pawar

Descriptive Statistics

Mean → average value
df["amount"].mean()

Median → central value
df["amount"].median()

Std dev → variation
df["amount"].std()

Quantiles → distribution cut
df["amount"].quantile([0.25,0.5,0.75])

Skew → distribution shape
df["amount"].skew()

Kurtosis → tail heaviness
df["amount"].kurt()

Mode → most frequent
df["status"].mode()

Pooja Pawar

GroupBy Analysis

Group mean → segment average
df.groupby("city")["amount"].mean()

Group sum → totals
df.groupby("city")["amount"].sum()

Multiple agg → deeper insight
df.groupby("city")
["amount"].agg(["mean","median","count"])

Pivot table → summary view
pd.pivot_table(df, values="amount", index="city")

Crosstab → category vs category
pd.crosstab(df["city"], df["status"])

Rank within group → comparison
df.groupby("city")["amount"].rank()

Top N per group → leaders
df.sort_values("amount").groupby("city").tail(3)

Pooja Pawar

Correlation & Relationships

Correlation matrix → relationships
df.select_dtypes("number").corr()

Target correlation → drivers
df.corr()["amount"]

Covariance → joint variation
df.cov()

Scatter plot → relation view
plt.scatter(df["x"], df["y"])

Pairplot → multi-feature view
sns.pairplot(df)

Heatmap → correlation visual
sns.heatmap(df.corr())

Line fit → trend check
np.polyfit(x, y, 1)

Pooja Pawar

Visual EDA

Histogram → distribution
df["amount"].hist()

Boxplot → outliers
df.boxplot(column="amount")

Bar plot → category counts
df["city"].value_counts().plot.bar()

Line plot → trends
df.plot.line(x="date", y="amount")

Countplot → frequency
sns.countplot(x="status", data=df)

Violin plot → density
sns.violinplot(x="status", y="amount", data=df)

Save plot → reuse
plt.savefig("plot.png")

Pooja Pawar

Automated EDA & Export

Profile report → full EDA
from ydata_profiling import ProfileReport
ProfileReport(df).to_file("eda.html")

Minimal profile → large data
ProfileReport(df, minimal=True)

Sweetviz → instant report
sv.analyze(df).show_html()

Missingno matrix → null pattern
msno.matrix(df)

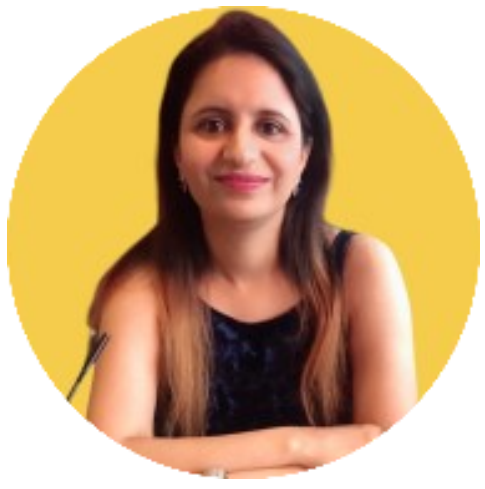
Export CSV → cleaned data
df.to_csv("clean.csv", index=False)

Export Parquet → analytics-ready
df.to_parquet("data.parquet")

Save stats → share insights
df.describe().to_csv("stats.csv")

Pooja Pawar

Follow for more data analytics content



Pooja Pawar