T3CH · Follow publication

Member-only story

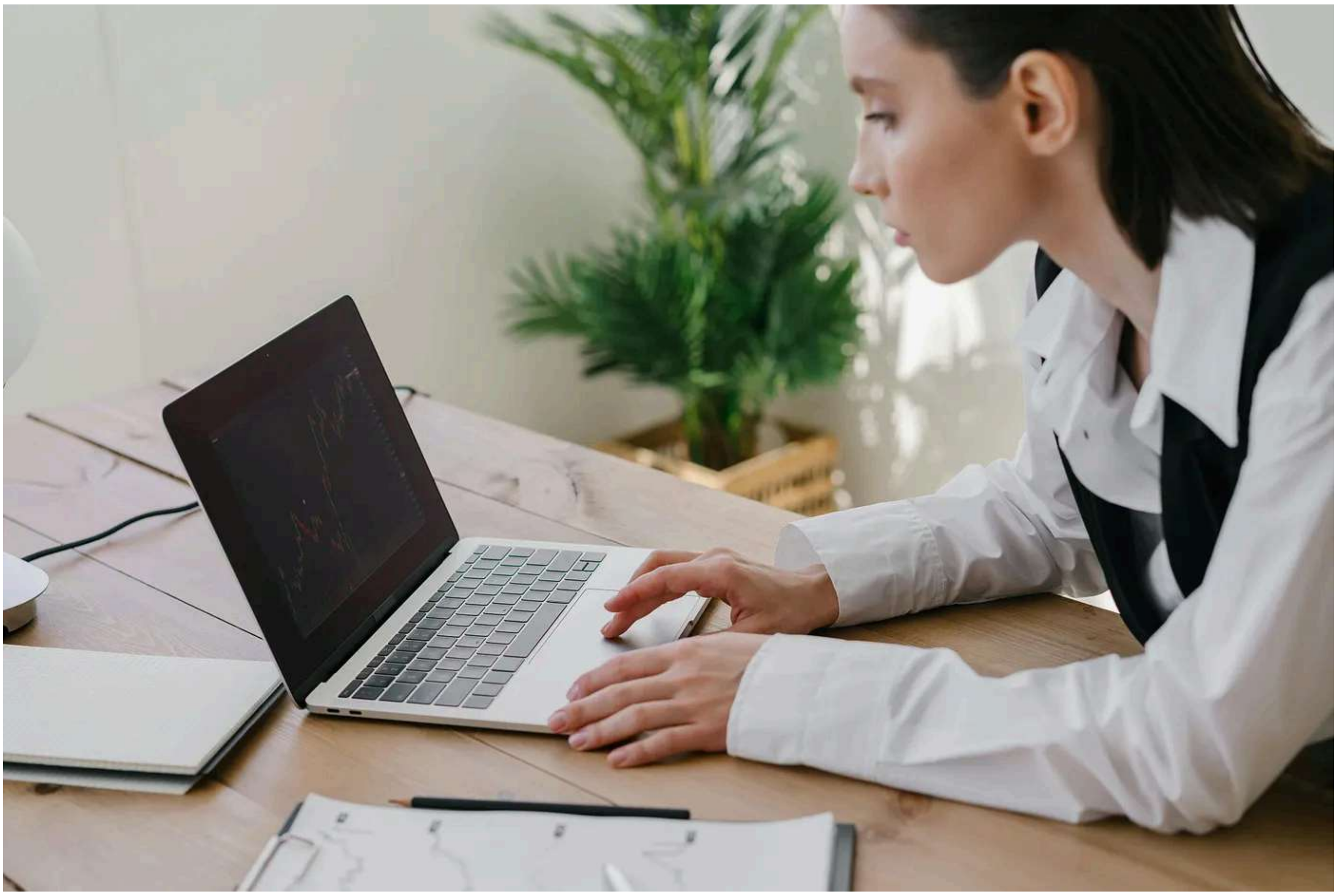
How to Perform Exploratory Data Analysis (EDA) for Better Insights

My Step-by-Step Journey from Raw Data to Clear Business Understanding.

3 min read · Sep 2, 2025

Maximilian Oliver Follow

Listen Share More



Whenever I start a new analytics project, I don't dive straight into machine learning or dashboards. Instead, I always begin with **Exploratory Data Analysis (EDA)**. EDA helps me **understand the story behind the data** — the trends, the outliers, the correlations, and sometimes the errors.

Here's how I typically perform EDA, broken into clear steps with practical examples and big code blocks.

1) Loading and Inspecting the Dataset

The first step is to **load the data and check its structure**.

```
import pandas as pd

# Load dataset
df = pd.read_csv("sales_data.csv")

# Preview
print(df.head())
print(df.info())
print(df.describe())
```

This gives me an idea of the **data types, missing values, and basic statistics**.

2) Handling Missing Data

Real-world data is never clean. Missing values are common.

```
# Count missing values
print(df.isnull().sum())

# Fill missing numerical columns with mean
```

```
df['Revenue'] = df['Revenue'].fillna(df['Revenue'].mean())

# Drop rows with too many missing values
df = df.dropna(thresh=3)
```

I usually decide between **imputing** or **dropping** depending on how important the column is.

3) Univariate Analysis

I always start by analyzing **one variable at a time.**

```
import matplotlib.pyplot as plt

# Histogram for Revenue
plt.hist(df['Revenue'], bins=30, edgecolor="black")
plt.title("Revenue Distribution")
plt.xlabel("Revenue")
plt.ylabel("Frequency")
plt.show()

# Value counts for categorical column
print(df['Region'].value_counts())
```

This helps me understand distributions and detect skewness or anomalies.

4) Bivariate Analysis

Next, I look at **relationships between two variables.**

```
import seaborn as sns

# Scatter plot: Revenue vs Marketing Spend
sns.scatterplot(x="Marketing_Spend", y="Revenue", data=df)
plt.title("Marketing Spend vs Revenue")
plt.show()
```

```
# Box plot: Revenue by Region
sns.boxplot(x="Region", y="Revenue", data=df)
plt.title("Revenue Distribution by Region")
plt.show()
```

I've often discovered **hidden patterns** here, like how certain regions outperform others.

5) Correlation Analysis

Correlation matrices help me see **how numerical features relate to each other.**

```
# Correlation matrix
corr = df.corr(numeric_only=True)

# Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

This step helps me identify **multicollinearity** or **predictive signals.**

6) Outlier Detection

Outliers can ruin analysis, so I always check them.

```
# Boxplot for outliers
sns.boxplot(x=df['Revenue'])
plt.title("Revenue Outlier Detection")
plt.show()

# Using IQR
Q1 = df['Revenue'].quantile(0.25)
Q3 = df['Revenue'].quantile(0.75)
```

```
IQR = Q3 - Q1
outliers = df[(df['Revenue'] < (Q1 - 1.5 * IQR)) | (df['Revenue'] > (Q3 + 1.5 *
print(outliers)
```

Depending on context, I either **remove or investigate outliers.**

Sometimes, I create new features to **enhance exploration.**

```
# Create new feature: Profit Margin
df['Profit_Margin'] = (df['Revenue'] - df['Cost']) / df['Revenue']

# Create new feature: Month from Date
df['Month'] = pd.to_datetime(df['Date']).dt.month

print(df[['Revenue', 'Profit_Margin', 'Month']].head())
```

This step often reveals **business-specific insights.**

8) Trend Analysis Over Time

For time series data, I look at **patterns over days, months, or years.**

```
df['Date'] = pd.to_datetime(df['Date'])
df = df.set_index('Date')

# Resample monthly
monthly_revenue = df['Revenue'].resample('M').sum()

monthly_revenue.plot(figsize=(10, 5))
plt.title("Monthly Revenue Trend")
```

```
plt.ylabel("Revenue")
plt.show()
```

This helps me detect **seasonality, growth, or decline**.

9) EDA Summary Report

After exploring, I summarize findings:

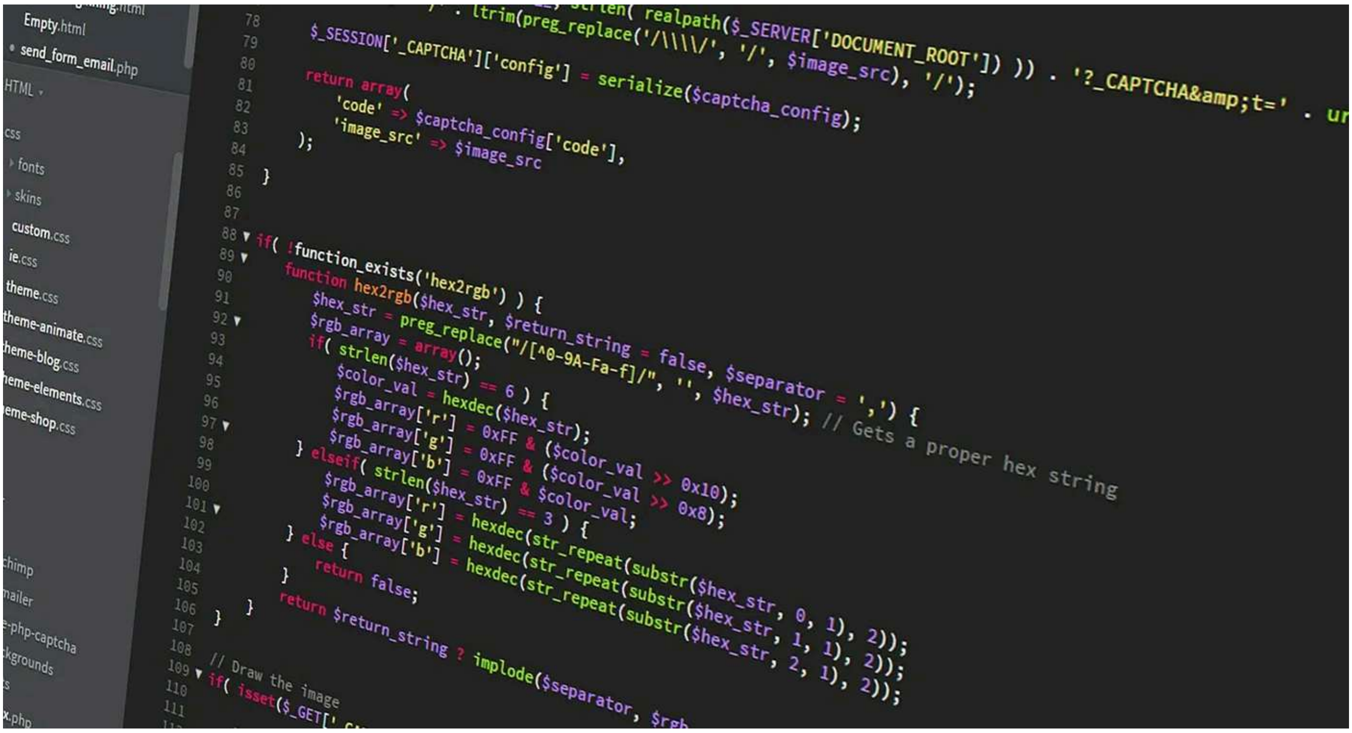
- Which variables are most important
- How revenue is distributed across regions and time
- Which marketing channels correlate most with sales
- What anomalies or errors exist

Sometimes, I even use **automated EDA tools** like `pandas-profiling` or `sweetviz`.

```
# pip install ydata-profiling
from ydata_profiling import ProfileReport

profile = ProfileReport(df, title="Sales Data EDA Report")
profile.to_file("eda_report.html")
```

This generates a **full interactive report** I can share with stakeholders.



Final Thoughts

EDA is the **foundation of all data projects**. Without it, you're flying blind. For me, EDA often reveals 70–80% of the insights I need before I even think about advanced models.

The key lesson: **don't rush into machine learning — first, let the data tell its story.**

Data Analysis · Data Analytics · Python Data Analysis · Eda · Data Insights

Follow

Published in T3CH

1.6K followers · Last published 2 days ago

Snoop & Learn about Technology, AI, Hacking, Coding, Software, News, Tools, Leaks, Bug Bounty, OSINT & Cybersecurity !i! But, not limited 2, anything that is Tech Linked...You'll probably find here ! ;)—Stay ahead with Latest Tech News! -> You write about? Just ping to join !