Project: Sales Data Analysis with NumPy, Pandas, and Matplotlib

In this project, you will work with a real-world sales dataset to perform end-to-end exploratory data analysis (EDA). You will use NumPy for numerical computations, Pandas for data manipulation, and Matplotlib for data visualisation. The goal is to understand the structure, trends, and relationships in the data and to generate meaningful business insights.

Dataset Description

The dataset `sales.csv` contains transactional sales data. Each record represents a product order with details such as order ID, customer, product, price, freight value, payment type, timestamps, and total price.

Column	**Description**
order_id	Unique order identifier.
customer_id	Unique customer key.
product_id	Unique product identifier.
category_english	Product category name.
price	Product price in local currency.
freight_value	Shipping cost for the order.
payment_type	Mode of payment (credit_card, boleto, etc.).
payment_value	Total payment amount for the order.
order_purchase_timestamp	Date and time when the order was placed.
order_delivered_customer_date	Date and time when the order was delivered.
total_price	Final amount charged (usually price + freight_value).

Project Tasks

Follow the steps below to complete your analysis. Write all code, outputs, and short observations for each task in your notebook. Ensure that your notebook is organised, with clear headings and labelled visualisations for every step.

- **Task 1:** Import the required libraries — NumPy, Pandas, and Matplotlib.
  Begin by importing the three core Python libraries needed for this project. These will help you perform numerical calculations, manipulate datasets, and create visualisations. Ensure that your environment is set up correctly before proceeding.
- **Task 2:** Load the dataset sales.csv using Pandas and display the first and last five rows.
  Read the dataset into a DataFrame and preview both the beginning and the end of the data to confirm it loaded correctly. Check that all columns are visible and that there are no formatting issues or unreadable values.

- **Task 3:** Check the number of rows and columns. Print basic info about the dataset using df.info().
  Explore the dataset structure by viewing its dimensions (row and column count) and the overall summary information, such as column names, data types, and non-null counts. This gives an initial understanding of data completeness.
- **Task 4:** Identify the data types of all columns and convert date columns to the datetime format.
  Review the type of data stored in each column (numeric, text, or date). Convert all date-related columns into proper datetime format so that you can later extract time-based features such as month or day of the week.
- **Task 5:** Check for missing values and duplicates. Handle them appropriately.
  Scan the dataset for any missing values and duplicate rows. Decide whether to fill missing values with substitutes (for example, average or median values) or remove them entirely. Eliminate duplicates to maintain clean data.
- **Task 6:** Use NumPy to calculate basic statistics — mean, median, min, max, and standard deviation for price and freight_value.
  Compute these key statistical measures to understand the central tendency and spread of product prices and freight values. Record these results clearly for comparison.
- **Task 7:** Create new derived columns such as delivery_days and total_cost (price + freight_value).
  Add calculated columns to your dataset: one showing the delivery time (difference between delivery and purchase dates) and another representing the total cost per order. These will be useful for deeper analysis later.
- **Task 8:** Extract additional date components — year, month, day, and day of week from purchase timestamp.
  Break down the purchase date into useful components such as year, month, day, and weekday. Store these as new columns so that you can explore time-based trends in later tasks.
- **Task 9:** Display the top 5 categories by total sales revenue.
  Group your data by product category and calculate total revenue for each. Identify the five categories that have generated the highest total sales and present them clearly, either as a table or a chart.
- **Task 10:** Find the most common payment type and visualise its proportion using a bar chart.
  Determine which payment method was used most frequently in the dataset. Create a bar chart that compares the frequency of all payment types to clearly show which one dominates.
- **Task 11:** Calculate average delivery time per category using groupby.
  Group your dataset by product category and calculate the average delivery time for each. Present the results so that it’s easy to identify which categories tend to deliver faster or slower.
- **Task 12:** Analyse total monthly sales trends — use line charts to visualise growth over time.

Aggregate your sales data by month and observe how total sales change over time. Create a line chart showing sales progression across months or years to reveal patterns or seasonality.

- **Task 13:** Create a histogram showing the distribution of product prices.
  Visualise how product prices are distributed within the dataset by creating a histogram. Label the axes appropriately to show how many products fall into each price range.
- **Task 14:** Use a box plot to compare freight_value across different categories.
  Plot the shipping cost distribution for each product category using a boxplot. This helps you compare variability and identify categories with unusually high or low shipping charges.
- **Task 15:** Identify the top 10 customers by total spending and visualise as a horizontal bar chart.
  Calculate how much each customer has spent in total and identify the top ten spenders. Create a horizontal bar chart ranking these customers from highest to lowest total spending for easy comparison.
- **Task 16:** Calculate the correlation between price, freight_value, and total_price. Visualise correlation using a heatmap.
  Measure how strongly these numerical variables are related to each other. Then, create a heatmap to visually display the correlation values in a clear and easy-to-read format.
- **Task 17:** Find how payment types vary by category — create a grouped bar plot.
  Compare the frequency of use of each payment method across different product categories. Create a grouped bar chart that displays this variation side by side for clear visual interpretation.
- **Task 18:** Using NumPy, create an array of total_price values and compute its percentile distribution (25th, 50th, 75th).
  Find the 25th, 50th (median), and 75th percentiles of total order values to understand how sales values are distributed across orders. Record and interpret the results briefly.
- **Task 19:** Display orders with exceptionally high total_price (e.g., above 95th percentile).
  Filter your dataset to show all orders whose total price exceeds the 95th percentile threshold. These represent outlier or high-value transactions that may warrant special attention.
- **Task 20:** Summarise key insights — list five business observations from your analysis.
  Write at least five key takeaways from your data exploration, focusing on patterns, trends, or findings that could be valuable for business decision-making. Keep your explanations concise but meaningful.
- **Task 21:** Identify the top 3 product categories that contribute the most to total revenue.
  Based on your revenue analysis, determine which three product categories contribute the highest share to total sales. Present these results in order of contribution.

- **Task 22:** Analyse if high shipping costs (freight_value) correlate with delayed deliveries.
  Compare shipping costs and delivery times to identify any possible relationship. Discuss whether higher shipping costs tend to reduce delivery time or if there's no clear link.
- **Task 23:** Find which months have the highest number of orders and discuss possible reasons (e.g., seasonal sales).
  Identify months with the most orders and explain potential reasons, such as festive seasons or sales periods, that might have influenced these peaks.
- **Task 24:** Compare average order value (AOV) between different payment types.
  Calculate the average value of orders for each payment method. Compare and interpret which payment types are associated with higher or lower order values.
- **Task 25:** Analyse customer purchase frequency — how many unique orders per customer exist? Identify the top 5 repeat customers.
  Determine how often each customer places orders by counting their unique transactions. Identify the five customers who have placed the most orders and discuss what this might indicate about customer loyalty or engagement.

Practice Guidelines

This week, you will use **Vocareum** to practice working with **Pandas** and **NumPy** for data analysis and transformation. You will load, inspect, clean, and transform a sales dataset step by step, reinforcing your skills in handling missing values, filtering, and creating new derived columns. You will also use **ChatGPT** as a support tool for debugging, clarifications, and exploring alternative approaches. No submission is required for this activity; the focus is on building fluency in data handling and preparation workflows.

To Use ChatGPT for This Assignment:

For this practice task, you can use the free web version of ChatGPT — no API or paid subscription is required.

- Go to https://chat.openai.com Links to an external site.in your browser.
- Log in using your email or Google/Microsoft account.
- Once inside, type your coding question or paste error messages directly into the chat box.
- Always review ChatGPT's suggestions. Don't copy blindly — test and adapt the code to your specific problem.

To access Vocareum:

- Go to the *Getting Started* module on your Course homepage (just below Programme Orientation).
- Click the Vocareum link (labelled Vocareum: Professional Certificate Programme in Agentic AI and Applications).
- Alternatively, you can also access it from the Course menu on the left side of your screen.

Note: Before reviewing any reference or solution set, make sure you attempt the practice independently. The solutions are meant for verification and deeper understanding only. Working through each step on your own will build lasting confidence and mastery of Pandas, NumPy, and Matplotlib for data analysis. This is a self-study activity for practice and does not count towards programme completion. No submission is required. However, completing the exercise on Vocareum will help you gain confidence in analysing and visualising data effectively.