A logo with a bird and a crown

AI-generated content may be incorrect.

**DATA CLEANING AND PREPARATION FOR E-COMMERCE ANALYSIS USING PYTHON IN JUPYTER NOTEBOOK: A CASE STUDY OF CUSTOMER AND TRANSACTIONAL DATASETS**

Contents

[1.0 Introduction 3](#_Toc195205863)

[1.1 Datasets Overview 3](#_Toc195205864)

[1.2 Data Cleansing Process 3](#_Toc195205865)

[2. Removing Duplicate Entries 7](#_Toc195205866)

[3. Standardizing Data Formats 7](#_Toc195205867)

[4. Summary of Cleansed Data 7](#_Toc195205868)

[4.1 Customer Demographics 7](#_Toc195205869)

[5. Tools Used 8](#_Toc195205870)

[6. Conclusion 8](#_Toc195205871)

[7. References 8](#_Toc195205872)

# 1.0 Introduction

Most data remain redundant until acted upon and processed to derive meaningful insights. Insights derived from data depend also on the quality of data, this is impacted by the data analysts’ ability to process data in a professional and unbiased manner to ensure the efficacy of data (Keller and Warrack, 2018).

Data cleaning is the first step in data analysis and management, it provides the building block on which meaningful insights are generated (Kotu and Deshpande, 2018). Cleaning data removes inconsistencies and **transform raw dataset** into **reliable resources.** It helps identify underlying issues in a dataset **like missing values, outliers, inconsistent formatting, duplicate records** which can make **for inaccuracy in analysis** and ultimately **decision-making.**

This report aims to highlight processes of data acquisition, describe the tools and step used in the data cleaning and present insight and summary statistics of the cleaned data making it ready for advanced analysis like trend revealing, customer behaviour among others. Effective data preparation is crucial for data analysis, as it lays the foundation for generating reliable and accurate insights. Without proper preparation, raw data can lead to misleading conclusions, biased results, or failed analyses. Proper data preparation ensures that data is clean, structured, and aligned with the specific objectives of the analysis (Kuhn and Johnson, 2013).

## 1.1 Datasets Overview

1. **Customers Dataset (customers.csv)**
   * Contains 30 records with the following columns: customer\_id, name, email, gender, age, and location.
   * Issues Identified: Missing values in email and age; inconsistent case formatting in location.
2. **Transactions Dataset (transactions.csv)**
   * Contains 45 records with the following columns: transaction\_id, customer\_id, product\_id, quantity, price, and transaction\_date.
   * Issues Identified: Missing values in quantity and price; inconsistent date formatting; duplicate entries.

## 1.2 Data Cleansing Process

**1. Identifying and Handling Missing Data**

The customers and transactions datasets exhibited missing values, which could skew analytical results if left unresolved. Missing data can be categorized as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR). For clarity, MCAR involves scenarios where the missing data is unrelated to any other variable in the dataset, MAR on the other hand includes when the missing values in one column can be accounted for given another variable in the dataset, whereas MNAR has to do with scenarios where the needed data was either not collected or not intentionally imputed. Understanding these makes for a best practice in technique used in mitigating their impact (Little and Rubin, 2019).

In the transaction’s dataset, **‘Price’** and **‘Quantity’** columns had missing data, while for Customers dataset, **‘Email’** and **‘Age’** had missing values. For this report, the missing values are assumed to be Missing Completely At Random (MCAR). The mean value for ‘Quantity’ and ‘Age’ columns were used to fill them up, while median value was used for ‘Price’. This was given the presence of outliers in the price field (minimum price of 7.99, maximum price of 129.99).

**Customers Dataset**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

##### Figure 1. Customer dataset cleaning codes

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

##### Figure 2. Transaction dataset cleaning

A screenshot of a computer code

AI-generated content may be incorrect.

A close-up of a logo

AI-generated content may be incorrect.

##### Figure 3. Pictorial representation of the cleaned customer data

# 2. Removing Duplicate Entries

The transactions dataset was scrutinized for duplicate rows based on identical transaction identifiers. A total of **1 duplicate row** was identified and removed, resulting in **44 unique transactions**. The customer dataset had no duplicate entry in it.

## 3. Standardizing Data Formats

Standardization of data formats is essential for ensuring consistency and reliability during analysis. Variability in data formats can cause errors during computation or hinder integration across different data sources (Batini and Scannapieco, 2016).

* **Date Formatting:** Transaction dates were converted to a consistent YYYY-MM-DD format to facilitate chronological analysis. The transaction\_date had different date formats that were not recognized as date at standardization, making for missing values in the column represented as ‘NaT’ (Not a Time). This called for linear method of interpolation to fill up the missing values and conversion to date to strip off the hours formatting (Judd, et al., 2020).
* **Numerical Data:** Numeric fields, such as quantity was converted to the appropriate data type, this could have been left as float depending on the values to ensure accuracy in computations. For the case of this analysis, it was converted to integer (as they were all whole numbers) (Murray and Johnson, 2018).
* **Categorical Date:** Categorical columns such as location was standardized, ensuring the consistent use of sentence casing. All text were converted to title case.

# 4. Summary of Cleansed Data

## 4.1 Customer Demographics

With the dataset ready for analysis, the cleansed customers.csv dataset was analysed to provide insights into the demographics of the customers. The following statistics were derived:

* **Age Distribution:** The ages of customers range from approximately 27 to 83 years, with a mean age of around 45years. There are fewer customers in the younger (27–33) and older age ranges (70–83). A majority of customers are clustered between **45 and 52**.
* **Gender Distribution:** Out of 30 customers, 14 (46.7%) are identified as female while 16 (53.3%) identified as male.
* **Location Breakdown:** The customers are spread across various locations. The most common locations include “Los Angeles” (30% of the total location), “New York” (23%) and “Chicago” (17%). These three locations make up for approximately 70% of the location transactions originated from.

These demographic insights provide a foundational understanding of the customer base, which can be useful for targeted marketing or segmentation analysis (Wedel and Kamakura, 2000).

* **Customers Dataset:** After cleansing, a total of **30 records** remained. Missing values for email and age were appropriately addressed.
* **Transactions Dataset:** After cleansing, a total of **44 unique transactions** were retained, free of duplicates and with standardized date formats.

# 5. Tools Used

For this cleaning exercise, only the Python’s Pandas library was imported, this is a very powerful module for data manipulation and analysis, there is the addition of infographics requiring the importation of another library Matplotlib. Functions such as .fillna(), .drop\_duplicates(), .dt.date(), pd.to\_datetime() and .astype() among others were utilized to treat missing data, remove duplicates, standardize formats, and convert data types effectively.

# 6. Conclusion

The cleaning process is an integral step in data analysis and management. This stage of data preparation is crucial for generating accurate, reliable insights that can drive informed decision-making (Zhu et al., 2019). The cleaning carried out above ensured no missing value exists in the datasets, removed duplicate entries and ensured consistency through standardization and formatting. Given this, advanced analytical processes can be carried out on the datasets.

# 7. References

Batini, C., & Scannapieco, M. (2016). Data and Information Quality: Dimensions, Principles and Techniques. Springer.

Judd, T., McIver, C., & Jacobs, M., 2020. *Data Cleaning Techniques: A Practical Approach to Handling Missing Data and Formatting Issues*. 2nd ed. Springer.

Keller, G. & Warrack, B., 2018. *Statistics for Business and Economics*. 9th ed. Cengage Learning.

Kotu, V., & Deshpande, B. (2019). *Data Science: Concepts and Practice*. Elsevier.

Kuhn, M. & Johnson, K., 2013. *Applied Predictive Modeling*. Springer.

Little, R. J. A., & Rubin, D. B. (2019). Statistical Analysis with Missing Data (3rd ed.). Wiley.

Murray, J. & Johnson, L., 2018. *Data Transformation and Management for Analysis*. 3rd ed. Wiley.

Rahm, E., & Do, H. H. (2000). Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin, 23(4), 3-13.

Wedel, M. and Kamakura, W.A., 2000. Market Segmentation: Conceptual and Methodological Foundations. Kluwer Academic Publishers.

Zhu, X., Wu, X., & Zhang, Z. (2019). Data Cleaning and Data Integration: A Comprehensive Survey. IEEE Transactions on Knowledge and Data Engineering, 31(9), 1584-1609.

A logo with a bird and a crown

AI-generated content may be incorrect.

**TITLE:**

**DESIGNING AN OPTIMIZED NOSQL SCHEMA FOR E-COMMERCE TRANSACTIONS AND PRODUCTS USING MONGODB GRAPHICS USER INTERFACE (GUI)**

Contents

[1. Introduction 3](#_Toc195213723)

[2. Database Overview and Tools Used 3](#_Toc195213724)

[2.1. Database Import Using MongoDB Compass GUI 4](#_Toc195213725)

[2.2. Database Set up 4](#_Toc195213726)

[2.3 Products Collection Import Using Compass: 4](#_Toc195213727)

[2.4 Transactions Collection Import 4](#_Toc195213728)

[3. Schema Design 4](#_Toc195213729)

[3.1 Design Requirements 4](#_Toc195213730)

[3.2 Schema Design 5](#_Toc195213731)

[3.3 Chosen Approach: Referencing vs Embedding 5](#_Toc195213732)

[4. Query Implementation and Results 6](#_Toc195213733)

[4.1 Navigating the Aggregations Tab 6](#_Toc195213734)

[4.2 Query 1 6](#_Toc195213735)

[4.3 Query 2 7](#_Toc195213736)

[4.4 Query 3 9](#_Toc195213737)

[4.5 Final Steps 9](#_Toc195213738)

[5. Evaluation and Discussion 10](#_Toc195213739)

[5.1 Effectiveness of Schema 10](#_Toc195213740)

[5.2 Query Relevance and Accuracy 10](#_Toc195213741)

[5.3 GUI vs CLI 11](#_Toc195213742)

[6. Conclusion 11](#_Toc195213743)

[7. References 11](#_Toc195213744)

# 1. Introduction

In today's data-driven landscape, NoSQL databases such as MongoDB have become instrumental in managing the complex, high-volume, and semi-structured datasets often encountered in e-commerce. This report explores the creation and implementation of a MongoDB-based database for Shopmart, focusing on transactions and product data. The project encompasses schema design, data import using MongoDB Compass Graphics User Interface (GUI), query generation, and rationale for design decisions. This work highlights how MongoDB can be leveraged to support scalable, effective, and flexible data storage and retrieval for an e-commerce setting.

In the era of big data and evolving digital markets, the demand for scalable, efficient, and flexible data storage solutions has led to the increasing adoption of NoSQL databases (Sadalage and Fowler, 2012). Among these, MongoDB stands out as a document-based NoSQL system that facilitates the efficient handling of unstructured and semi-structured data (Moniruzzaman and Hossain, 2013). Unlike relational databases, which rely on rigid schema definitions and table relationships, MongoDB offers dynamic schemas that are highly adaptable to the needs of modern e-commerce platforms like Shopmart (Banker, 2011).

This report details the development of a MongoDB database schema for Shopmart, an online retail platform. It includes the rationale for design decisions, the steps taken to set up the database and import data, and the implementation of key queries that align with the business objectives using MongoDB Compass GUI. The core goal is to create a schema that not only supports current data management needs but also scales as the business grows. Throughout this report, MongoDB’s document-oriented capabilities are leveraged to meet common e-commerce use cases such as tracking customer transactions, evaluating product performance, and managing inventory efficiently. This work highlights how MongoDB can be leveraged to support scalable, effective, and flexible data storage and retrieval for an e-commerce setting.

# 2. Database Overview and Tools Used

The MongoDB database created for Shopmart is named shopmart.db. The implementation utilises the MongoDB Compass GUI to enhance user experience, reduce command-line complexity, and allow visual exploration of data structures. GUI-based management is especially beneficial in academic and business contexts, where users may not have strong technical backgrounds (Chen, et al., 2020).

## 2.1. Database Import Using MongoDB Compass GUI

GUI import avoids manual formatting issues common in CLI operations (MongoDB Inc., 2023). The cleaned transaction and product data, previously processed to handle inconsistencies such as duplicates and formatting issues, is imported into MongoDB using the Compass interface. This ensures a smooth transition from CSV format to BSON, which MongoDB uses internally.

## 2.2. Database Set up

• Launch MongoDB Compass on your local computer and click Connect from the left-hand sidebar (LHS).

• Once connected, the default system databases—**admin**, **config**, and **local**—which are used for system-level operations displays. These should not be modified unless by an administrator.

• From the LHS, click Create Database.

• Enter the database name *Shopmart* in the dialog box and click *Enter* to create the database.

## 2.3 Products Collection Import Using Compass:

* Select the shopmart.db database.
* Create a new collection named Products.
* Click "Add Data" > "Import File" > choose products.csv.
* Map fields accordingly (e.g., product\_id as Integer, price as Number).

## 2.4 Transactions Collection Import

* Create a new collection named Transactions.
* Repeat the import steps with transactions.csv.
* Ensure date fields are correctly formatted

# 3. Schema Design

## 3.1 Design Requirements

The schema is designed to answer the following:

1. Track customer purchases by product category
2. Identify popular products based on sales volume
3. Monitor stock levels of frequently purchased items

## 3.2 Schema Design

The schema is organised into two main collections: Transactions and Products. The Transactions collection records individual purchases made by customers, while the Products collection stores information about the items being sold.

**Transactions Collection:**

* \_Id: ObjectId
* transaction\_id: Integer (Unique identifier for each transaction)
* customer\_id: Integer (Reference to the customer making the purchase)
* product\_id: Integer (Reference to the purchased product)
* quantity: Integer
* price: Double
* transaction\_date: Date (Format: YYYY-MM-DD)

**Products Collection:**

* \_Id: ObjectId
* product\_id: Integer (Unique identifier for each product)
* product\_name: String
* category: String
* price: Double
* stock: Integer

## 3.3 Chosen Approach: Referencing vs Embedding

MongoDB offers two primary approaches for schema design: referencing and embedding. Embedding entails storing nested documents, while referencing links collections using ObjectIds or custom identifiers (Sadalage and Fowler, 2012).

For Shopmart, the **referencing approach** is adopted—storing product IDs in the Transactions collection while maintaining a separate Products collection. This design decision is justified by several factors:

* **Data Redundancy Minimization**: Product data like price, category, and description should not be duplicated across multiple transaction records (Banker, et al., 2013).
* **Scalability**: Referencing ensures that product updates (e.g., price changes) are reflected globally without re-writing all transaction records.
* **Separation of Concerns**: Transactions and products are logically distinct entities, and referencing respects their modularity (Bradshaw, Brazil and Chodorow, 2019).
* **Ease of Updates**: With referencing, updates to product information (such as price changes or category updates) can be made in one location. This avoids inconsistencies that can arise from embedding.

The embedded array items within each transaction allows for flexibility in the number of products purchased per transaction, while the product\_id acts as a foreign key to the Products collection.

The trade-off with referencing is the need for joins to combine data from multiple collections, which can increase query complexity and reduce read performance(Roffe, 2024). Nevertheless, MongoDB documents have a size limit of 16 MB, which can be restrictive for large datasets making referencing approach a better alternative. (MongoDB Inc., 2023).

# 4. Query Implementation and Results

**All queries were run through Compass's Aggregation Pipeline builder.**

## 4.1 Navigating the Aggregations Tab

1. In Compass, click on your **Transactions** collection.
2. Select the **"Aggregations"** tab at the top.
3. Click **“Add Stage”** to begin constructing your queries.

## 4.2 Query 1

**Top 5 Products by Total Sales**

This aggregation calculates revenue per product using price \* quantity and sorts by total revenue.

A screenshot of a computer

AI-generated content may be incorrect.

A white rectangular object with a white background

AI-generated content may be incorrect.

A white rectangular object with a black border

AI-generated content may be incorrect.

**Figure 1a. Top 5 products by sales query**

**Output**

A screenshot of a cell phone

AI-generated content may be incorrect.

**Figure 1b. Top 5 products by sales output**

## 4.3 Query 2

**Highest Sales Volume by Category**

This query joins Transactions with Products to access category information, then sums quantities sold per category.

A screenshot of a computer

AI-generated content may be incorrect.

**A white rectangular object with a black border

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A white screen with a black text

AI-generated content may be incorrect.**

**Figure 2a. Highest sales volume by category query**

**Output**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 2b. Highest sales volume by category output**

## 4.4 Query 3

**Average Transaction Amount**

The goal of this query is to calculate the average value of transactions across all data.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 3a. Average transaction amount query**

**Output**

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 3b. Average transaction amount output**

## 4.5 Final Steps

Once you've executed each query:

* Review the **Preview Results** in Compass.
* Click **"Export Pipeline"** to save the queries in JSON or a programming language format.
* Use **"Export Data"** to download query results.

# 5. Evaluation and Discussion

## 5.1 Effectiveness of Schema

The referencing model provides a robust and scalable structure. It aligns with normalisation principles, reducing data redundancy while supporting updates across interrelated datasets (Kaur and Rani, 2021). Embedding would have increased redundancy, especially for popular products appearing in many transactions. A robust schema should not only support existing data needs but also accommodate future growth (MongoDB Inc., 2023). The referencing model used in this design aligns with MongoDB best practices for many-to-one relationships (Zola, 2022). While $lookup introduces some performance overhead compared to embedded models, it offers greater flexibility and adheres to sound data management principles (Roffe, 2024).

This design enables efficient querying for core e-commerce metrics, such as identifying top-selling products and computing category-level insights. From an analytical perspective, this structure supports modular querying, allowing the business to perform customer segmentation, sales forecasting, and inventory planning (Chen, Chian and Storey, 2020).

Moreover, the database can be scaled horizontally using MongoDB’s sharding capabilities, ensuring that Shopmart remains responsive as its customer base and transaction volume grow (MongoDB, n.d.). Indexing on key fields such as transaction\_id, product\_id, and category can further enhance performance.

The modularity of this schema also benefits operational tasks such as product restocking and catalogue updates. For example, product stock levels can be adjusted independently of transaction data, and new categories can be introduced without modifying historical records.

## 5.2 Query Relevance and Accuracy

The queries were designed to extract actionable insights:

* Query 1 identifies best-performing products
* Query 2 evaluates category-level sales trends
* Query 3 provides macro-level transaction insights

These analytics can guide business strategies in inventory management, marketing, and customer targeting.

## 5.3 GUI vs CLI

Using the GUI provides visual insights into schema validation, data typing, and real-time query construction. For non-technical users, GUI simplifies complex database interactions, aligning with best practices in user-centered design (Oussous, et al., 2018).

# 6. Conclusion

This report presents a complete lifecycle of developing a MongoDB database for an e-commerce platform using GUI tools. The use of referencing in schema design demonstrated an optimal balance between performance, scalability, and maintainability. It efficiently supports typical e-commerce use cases such as tracking transactions, identifying popular products, and analysing category performance. While embedding could offer faster reads, referencing offers flexibility and scalability that are essential for long-term growth and data integrity. As such, this design provides a strong foundation for ShopMart’s data infrastructure and future analytical needs.

GUI-based interactions ensured accessibility, making the system viable for users with diverse technical backgrounds. Through thoughtful queries and data modelling, this implementation supports actionable business insights and sets a foundation for further analytics development.

# 7. References

* Banker, K. (2011). *MongoDB in Action*. Manning Publications.
* Banker, R. D., Bardhan, I. R., & Asdemir, O. (2013). Understanding the impact of collaboration software on product design and development. Information Systems Research, 24(4), 1091-1107.
* Bradshaw, S., Brazil, E. and Chodorow, K., 2019. *MongoDB: the definitive guide: powerful and scalable data storage*. " O'Reilly Media, Inc.".
* Chen, H., Chiang, R. H. L., & Storey, V. C. (2020). Business intelligence and analytics: From big data to big impact. MIS Quarterly, 36(4), 1165-1188.
* Kaur, K., & Rani, R. (2021). Modeling and Querying Data in NoSQL and NewSQL Databases. In Big Data and NoSQL (pp. 73-93). Springer.
* MongoDB (no date) *Sharding in MongoDB*. Available at: <https://www.mongodb.com> (Accessed: 9 April 2025).
* MongoDB Inc. (2023) 'Data modeling', MongoDB Manual v8.0. Available at: [MongoDB Documentation](https://www.mongodb.com/docs/manual/data-modeling/) (Accessed: 7 April 2025).
* MongoDB Inc. (2023) 'Import and export data with MongoDB Compass', MongoDB Manual v8.0. Available at: [MongoDB Documentation](https://www.mongodb.com/docs/compass/current/import-export/).
* MongoDB Inc. (2023) 'MongoDB limits and thresholds', MongoDB Manual v8.0. Available at: [MongoDB Documentation](https://www.mongodb.com/docs/manual/reference/limits/) (Accessed: 7 April 2025).
* MongoDB Inc. (2023) 'Embedded data versus references', Database Manual v8.0. Available at: [MongoDB Documentation](https://www.mongodb.com/docs/manual/data-modeling/concepts/embedding-vs-references/) (Accessed: 7 April 2025).
* Moniruzzaman, A.B.M. and Hossain, S.A., 2013. *NoSQL database: New era of databases for big data analytics - classification, characteristics and comparison*. International Journal of Database Theory and Application, 6(4), pp.1-14. Available at: <https://doi.org/10.14257/ijdta.2013.6.4.01>
* Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University - Computer and Information Sciences, 30(4), 431-448.
* Roffe, S. (2024) 'Optimizing database joins in NoSQL environments', *Journal of Database Systems*, 35(2), pp. 145–160. doi: 10.1109/JDS.2024.00160
* Roffe, S. (2024) 'Optimizing $lookup performance using the power of indexing', MongoDB Developer Center. Available at: [MongoDB Developer Center](https://www.mongodb.com/developer/products/mongodb/optimizing-lookup-performance-using-power-of-indexing/) (Accessed: 7 April 2025).
* Sadalage, P. J., & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley.
* Schoemaker, P. J. (1995). Scenario planning: A tool for strategic thinking. Sloan Management Review, 36(2), 25.
* Zola, W. (2022) '6 rules of thumb for MongoDB schema design', MongoDB Blog. Available at: [MongoDB Blog](https://www.mongodb.com/blog/post/6-rules-of-thumb-for-mongodb-schema-design) (Accessed: 7 April 2025).

A logo with a bird and a crown

AI-generated content may be incorrect.

**REPORT ON POWER BI DASHBOARD CREATION**

**AN ANALYSIS OF SHOPMART DATABASE FOR DECISION MAKING**

**Introduction**

This dashboard is designed to empower sales managers and business analysts with actionable insights for strategic decision-making. By presenting clear, focused visualizations, it enables users to monitor sales trends, evaluate category performance, and identify opportunities for improvement. The design prioritizes usability, clarity, and relevance to support data-driven business decisions.

A screenshot of a dashboard

AI-generated content may be incorrect.

**Figure 1. Shopmart data analysis visualization**

**Target Audience**

The dashboard is tailored for **sales managers** and **business analysts** who are responsible for monitoring performance and making strategic decisions. These professionals need accessible, actionable insights to optimize sales operations, forecast trends, and identify areas of improvement (Few, 2006).

**How Each Visualization Supports Decision-Making**

1. **Line Chart for Sales Trends**:
   * This chart tracks sales over time (year, quarter, month, and day). It allows users to identify seasonal patterns, predict future performance, and spot unexpected dips or spikes. The Line chart is most optimal for visualizing time-based data (Knaffic, 2015).
   * For example, a sudden drop in sales in a particular month could alert managers to operational issues like stock shortages or ineffective promotions. One can also see that the highest sales in the period was made in 2023, Quarter 1, in the 11th day of February.
2. **Bar Chart for Sales by Category**:
   * By visualizing sales distribution across product categories, this chart highlights which categories are driving revenue and which are underperforming by the proportion of sales each has. A pie chart might seem like a natural choice for proportion. However, with 9 categories, a pie chart becomes, **visually cluttered**, hard to **compare slice sizes** and **difficult to label,** making it less effective for comparison (Few, 2006).
   * Sales managers can use this insight to allocate resources efficiently **investing more** in **top-performing categories** (Devices) while **improving weaker ones** (Accessories).
3. **KPI Card (Average Sales 49.64)**

* The average sales KPI offers a quick reference to gauge overall sales health. KPI indicators provide high-level performance signals for faster decision-making (Yigitbasioglu and Velcu, 2012).

1. **Filters for Key Dimensions**:
   * Filters enable users to narrow down the data by **date range**, **product category**, and **customer segments** (e.g., age, gender, locations). This makes it possible to perform a drill-down analysis.
   * For instance, a manager could filter sales data to focus exclusively on high-value customers or analyse trends in a specific region.

For the visualization above, **page filters ({‘age’, ‘location’, ‘gender’, ‘category’, and ‘transaction\_date’})** were added to the filter pane, this ensures **interactivity** of all charts in the visualization.

**Rationale Behind Design Choices**

The layout uses a clean, minimalist design to reduce cognitive load. The visuals are positioned logically—trend on the left, categorical performance on the right, and a KPI metric at the top for instant insight. Consistent blue coloring across charts ensures visual coherence and ease of comparison.

1. **Simplicity**:
   * The dashboard uses clear, focused visuals and avoids clutter, ensuring that users can quickly understand the information without being overwhelmed (Few, 2006).
2. **Clarity**:
   * Each visualization serves a specific purpose, with intuitive filters to enhance interactivity. This makes the dashboard easy to navigate and actionable (Knaflic, 2015).
3. **Relevance**:
   * Every chart and filter is designed to align with the audience’s goals—helping sales managers and analysts make informed decisions quickly, such as tweaking strategies or reallocating resources.

**References**

Few, S. (2006) *Information Dashboard Design:* *The Effective Visual Communication of Data.* Sebastopol, CA: O'Reilly Media.

Few, S. (2009) *Now You See It:* *Simple Visualization Techniques for Quantitative Analysis.* Oakland, CA: Analytics Press.

Knaflic, C.N. (2015) *Storytelling with Data: A Data Visualization Guide for Business Professionals.* Hoboken, NJ: Wiley.

Shneiderman, B. (1996) ‘The eyes have it: A task by data type taxonomy for information visualizations’, *Proceedings 1996 IEEE Symposium on Visual Languages*, Boulder, CO, USA, 3–6 September. IEEE, pp. 336–343. doi:10.1109/VL.1996.545307.

Yigitbasioglu, O.M. and Velcu, O. (2012) ‘A review of dashboards in performance management: Implications for design and research’, *International Journal of Accounting Information Systems,* 13(1), pp. 41–59. doi:10.1016/j.accinf.2011.08.002.