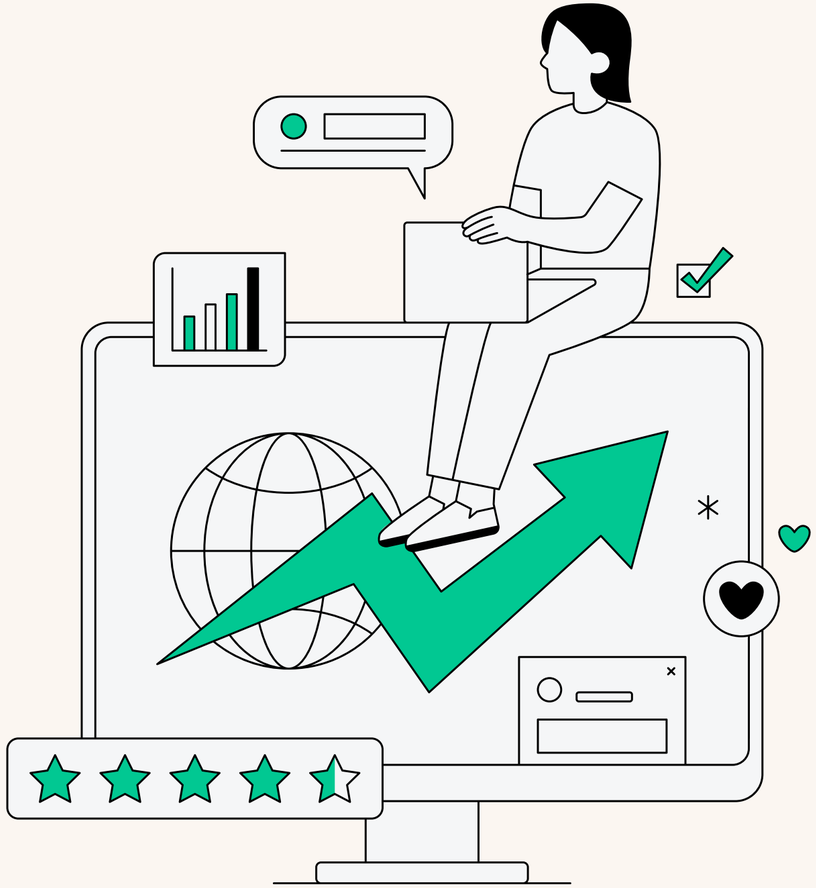
Data Analysis final project

Presented by **Francesco Genna**



Introduction

Imagine working for a second-hand clothing marketplace, a platform promoting sustainable fashion.

The marketplace offers people an easy and safe way to resell their used clothes, giving clothes a second life and promoting a more sustainable lifestyle.

Here, anyone can offer clothes, accessories and footwear for sale, allowing others to find unique pieces at affordable prices👗

The platform is designed for people who love fashion but also want to reduce their environmental impact, offering an ethical alternative to fast fashion.
Every purchase helps reduce waste and promotes a circular economy.

Dataset

- **user_uuid** : Identifier of the selling user
- **category** : Category of clothing
- **designer_id** : Identifier of the designer
- **language**: Language of the selling user
- **level**: Price range
- **country**: Nationality of the selling user
- **purchase_date** : The date on which the user sold the product
- **platform:** Platform from which the payment was made
- **item_id**: Product identifier
- **stars**: Average stars assigned to the product (1 to 5)
- **subscription_date**: Day on which the selling user subscribed

To visualize the entire dataset, click on this link

Work Pipeline

EDA
Exploratory data analysis

Exploration and general understanding, as well as visualization of data, using *Python* and its libraries Pandas, Matplotlib, and Seaborn.

Visual exploration

Data cleaning and wrangling

Correction of data types, removal of unnecessary columns, modification of values to make them more explanatory, and creation of new columns.

Data manipulation and modeling

Process the data and create structured and meaningful tables. This phase was realised by integrating Python and PostgreSQL, thus facilitating more in-depth and targeted analyses

Visualization and insights

The data visualization was carried out using Tableau to create interactive and effective dashboards, aimed at clearly communicating the insights gained from the analysis.

EDA - Exploratatory Data Analysis

```
fashion_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   user_uuid          999 non-null    int64 
 1   category           999 non-null    object
 2   designer_id        999 non-null    int64 
 3   language           999 non-null    object
 4   level              999 non-null    object
 5   country            999 non-null    object
 6   purchase_date      999 non-null    object
 7   platform           999 non-null    object
 8   item_id            999 non-null    int64 
 9   stars              999 non-null    int64 
 10  subscription_date  999 non-null    object
```

```
fashion_data.describe()
```

	purchase_date	purchase_year	purchase_month	stars	subscription_date	subscription_year	subscription_month
count	999	999.000000	999.000000	999.000000	999	999.000000	999.000000
mean	2023-01-01 10:52:58.378378240	2022.505506	6.532533	3.013013	2021-06-22 10:11:10.270270208	2020.997998	6.277277
min	2021-01-01 00:00:00	2021.000000	1.000000	1.000000	2020-01-02 00:00:00	2020.000000	1.000000
25%	2022-06-26 00:00:00	2022.000000	4.000000	2.000000	2020-12-01 00:00:00	2020.000000	3.000000
50%	2023-02-26 00:00:00	2023.000000	7.000000	3.000000	2021-06-05 00:00:00	2021.000000	6.000000
75%	2023-07-28 00:00:00	2023.000000	9.000000	4.000000	2022-02-18 00:00:00	2022.000000	9.000000
max	2023-12-28 00:00:00	2023.000000	12.000000	5.000000	2022-12-23 00:00:00	2022.000000	12.000000
std	NaN	0.660752	3.483527	1.415216	NaN	0.764962	3.561961

```
fashion_data.duplicated().sum()
```

```
0
```

- Using the .info() function, we can observe that there are no null values in the dataset.
- Some data types are incorrect, for example the purchase and subscription dates
- Using the .describe() method, it can be observed that the Stars column shows a fairly symmetric distribution.
- To check for duplicate rows, the .duplicated().sum() method was used, which returns 0.

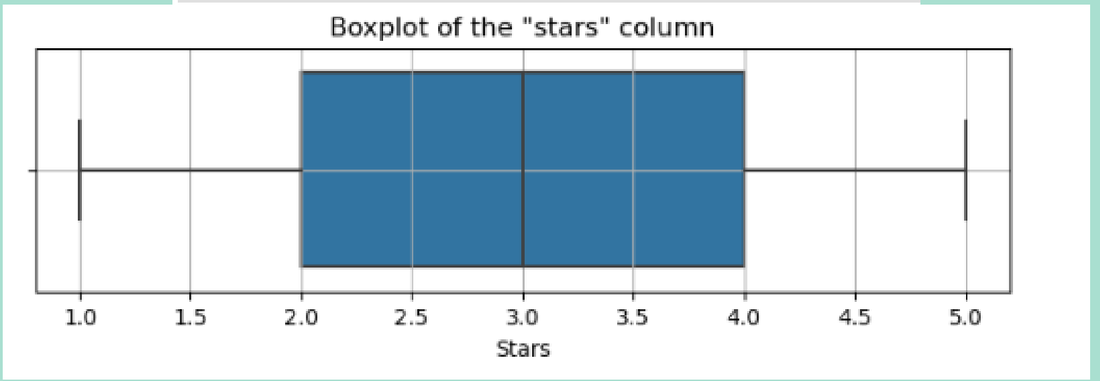
EDA - Exploratatory Data Analysis

Statistical description of Stars column (rating)

```
fashion_data["stars"].describe()
```

```
count    999.000000
mean       3.013013
std        1.415216
min        1.000000
25%        2.000000
50%        3.000000
75%        4.000000
max        5.000000
Name: stars, dtype: float64
```

```
import matplotlib.pyplot as plt
import seaborn as sns


# Boxplot to display outliers and median
plt.figure(figsize=(8, 2))
sns.boxplot(x=fashion_data['stars'])
plt.title('Boxplot of the "stars" column')
plt.xlabel('Stars')
plt.grid(True)
plt.show()
```



The stars column shows a symmetrical distribution, with no outliers

Data cleaning and wrangling

Conversion of date columns into datetime format and transformation of some columns into strings
purchase_date' and “subscription_date” are converted from strings to datetime with day-month-year format, using ‘coerce’ to handle invalid values such as NaT
user_uuid', “designer_id” and “item_id” are converted to strings to ensure data type consistency

```
fashion_data['purchase_date'] = pd.to_datetime(fashion_data['purchase_date'], format='%d-%m-%Y', errors='coerce')
fashion_data['subscription_date'] = pd.to_datetime(fashion_data['subscription_date'], format='%d-%m-%Y', errors='coerce')
fashion_data['user_uuid'] = fashion_data['user_uuid'].astype(str)
fashion_data['designer_id'] = fashion_data['designer_id'].astype(str)
fashion_data['item_id'] = fashion_data['item_id'].astype(str)
```

Contingency table between language and country

```
# Contingency table between language and country
cross_tab = pd.crosstab(fashion_data['language'], fashion_data['country'])

print(cross_tab)
```

```
country     fr   it   uk
language
en           0    0  337
fr         211    0    0
it           0  451    0
```

The language column does not provide additional insight, as it perfectly correlates with the country column.
Therefore the language column can be excluded.

Data cleaning and wrangling

```python
def assign_trimester(month):
    """
    Assigns a quarter (T1-T4) based on the month.

    Parameters:
    month (int): Mese as integer (1-12)

    Returns:
    str: Trimester as 'T1', 'T2', 'T3' o 'T4'
    """
    if month in [1, 2, 3]:
        return 'T1'
    elif month in [4, 5, 6]:
        return 'T2'
    elif month in [7, 8, 9]:
        return 'T3'
    else:
        return 'T4'

# Extract year, quarter and month from purchase_date
fashion_data['purchase_year'] = fashion_data['purchase_date'].dt.year
fashion_data['purchase_trimester'] = fashion_data['purchase_date'].dt.month.apply(assign_trimester)
fashion_data['purchase_month'] = fashion_data['purchase_date'].dt.month

# Extract year, quarter and month from subscription_date
fashion_data['subscription_year'] = fashion_data['subscription_date'].dt.year
fashion_data['subscription_trimester'] = fashion_data['subscription_date'].dt.month.apply(assign_trimester)
fashion_data['subscription_month'] = fashion_data['subscription_date'].dt.month
```

Break down the column of purchase dates and subscription date.

This helps to analyse seasonal and annual purchasing trends, useful for marketing strategies, planning and identifying sales peaks.

```python
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")

fig, axes = plt.subplots(1, 2, figsize=(16, 6))
fig.suptitle('Visualization Analysis', fontsize=18)

# 1. Total number of transactions per category (barplot)
transactions_per_category = fashion_data['category'].value_counts().reset_index()
transactions_per_category.columns = ['category', 'transaction_count']
transactions_per_category = transactions_per_category.sort_values(by='transaction_count', ascending=True)

sns.barplot(data=transactions_per_category, x='category', y='transaction_count', palette='viridis', ax=axes[0])
axes[0].set_title('Total Transactions per Category')
axes[0].set_xlabel('Category')
axes[0].set_ylabel('Transaction Count')
axes[0].tick_params(axis='x', rotation=45)

# 2. Average rating (stars) per category (barplot)
avg_stars = fashion_data.groupby('category')['stars'].mean().reset_index()
avg_stars = avg_stars.sort_values(by='stars', ascending=True)

sns.barplot(data=avg_stars, x='category', y='stars', ax=axes[1], palette='flare')
axes[1].set_title('Average Rating per Category')
axes[1].set_xlabel('Category')
axes[1].set_ylabel('Average Stars')
axes[1].tick_params(axis='x', rotation=45)

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

# 3. Average rating per price level (barplot)
plt.figure(figsize=(8, 5.5))
level_stars = fashion_data.groupby('level')['stars'].mean().reset_index()
sns.barplot(data=level_stars, x='level', y='stars', palette='light:#5A9', width=0.5)
plt.title('Average Rating per Price Level')
plt.xlabel('Level')
plt.ylabel('Average Stars')
plt.ylim(1, 5)
plt.tight_layout()
plt.show()

# 4. Platform distribution (pie chart)
platform_counts = fashion_data['platform'].value_counts()
platform_labels = platform_counts.index
platform_sizes = platform_counts.values

plt.figure(figsize=(5.5, 5.5))
plt.pie(platform_sizes, labels=platform_labels, autopct='%1.1f%%', startangle=140,
        colors=sns.color_palette('pastel'))
plt.title('Platform Distribution (%)')
plt.axis('equal')
plt.tight_layout()
plt.show()

# 5. Number of items per price level (barplot)
plt.figure(figsize=(8, 5.5))
item_counts_per_level = fashion_data.groupby('level')['item_id'].count().reset_index().rename(columns={'item_id': 'item_count'})
sns.barplot(data=item_counts_per_level, x='level', y='item_count', palette='Blues_d')
plt.title('Number of Items per Price Level')
plt.xlabel('Level')
plt.ylabel('Item Count')
plt.tight_layout()
plt.show()
```

Visual exploration

This code performs an **exploratory visual analysis** of the fashion_data dataset using the visualization libreries *Seaborn* and *Matplotlib*.

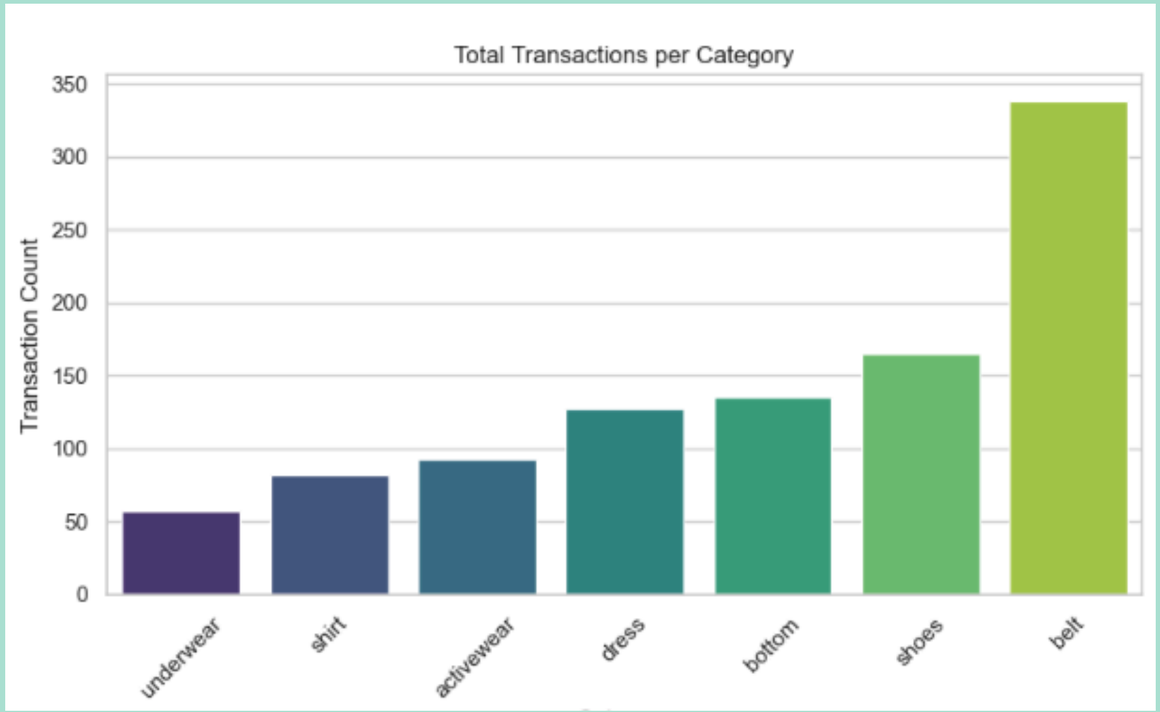
It shows the number of unique users and the average rating per category, the average star rating per price level, the percentage distribution of platforms via a pie chart and the number of items per level.

The objective is to identify patterns in purchasing and rating behaviour, which is useful for understanding user preferences, category performance and platform utilisation.

* Due to exposure issues, the code for generating the last graph is not present, but is visible on the notebook

The graphs are shown on the next slides

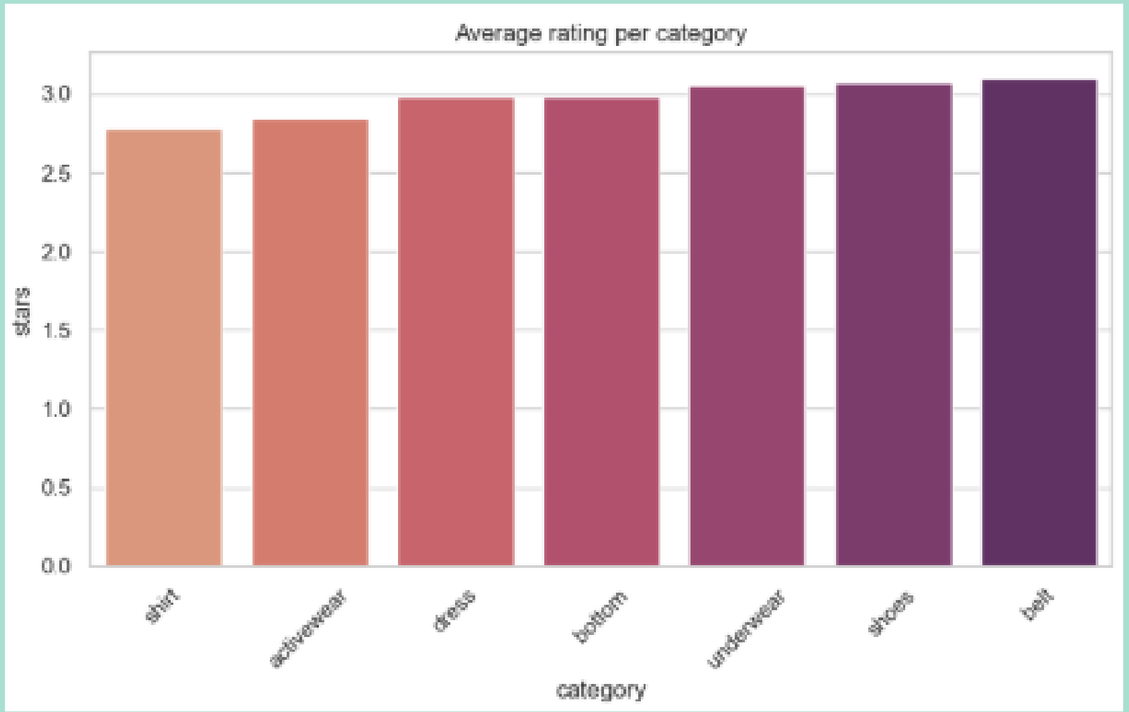
Visual exploration
Category



X-axis : categories.

Y-axis: the number of transactions.

The 'belt' category has the highest number of transactions.



X-axis : categories.

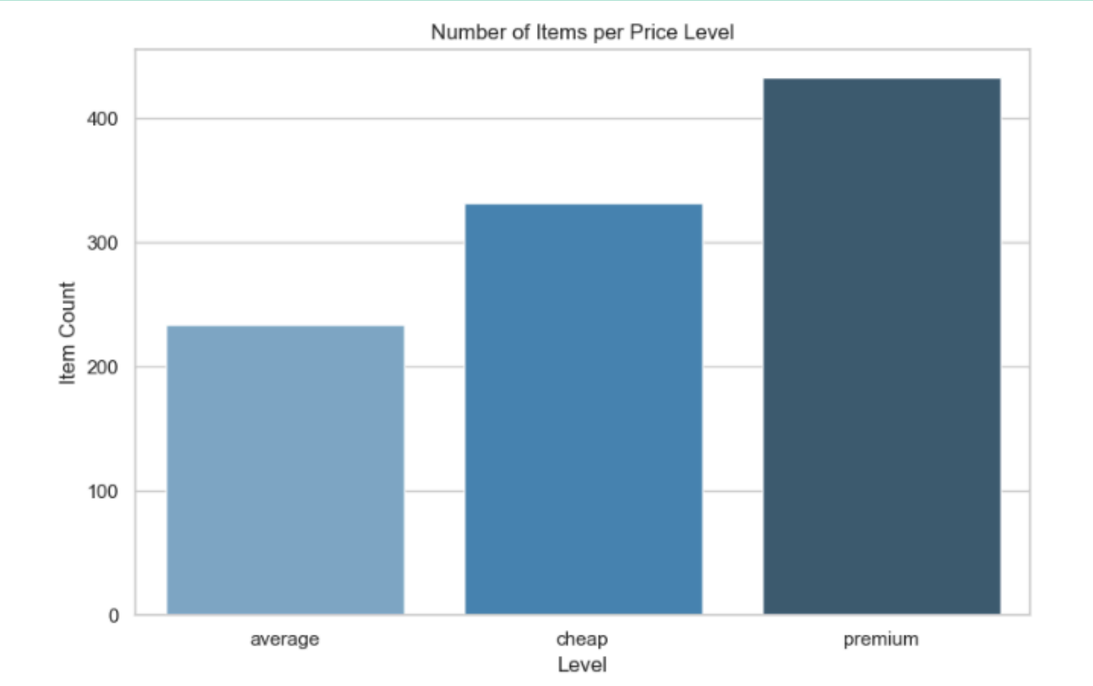
Y-axis: average stars (rating).

The 'Belt' category has the highest average rating, but it also has the highest number of transactions.

In contrast, the 'Underwear' category has the third highest average rating, despite having the lowest number of transactions

Visual exploration

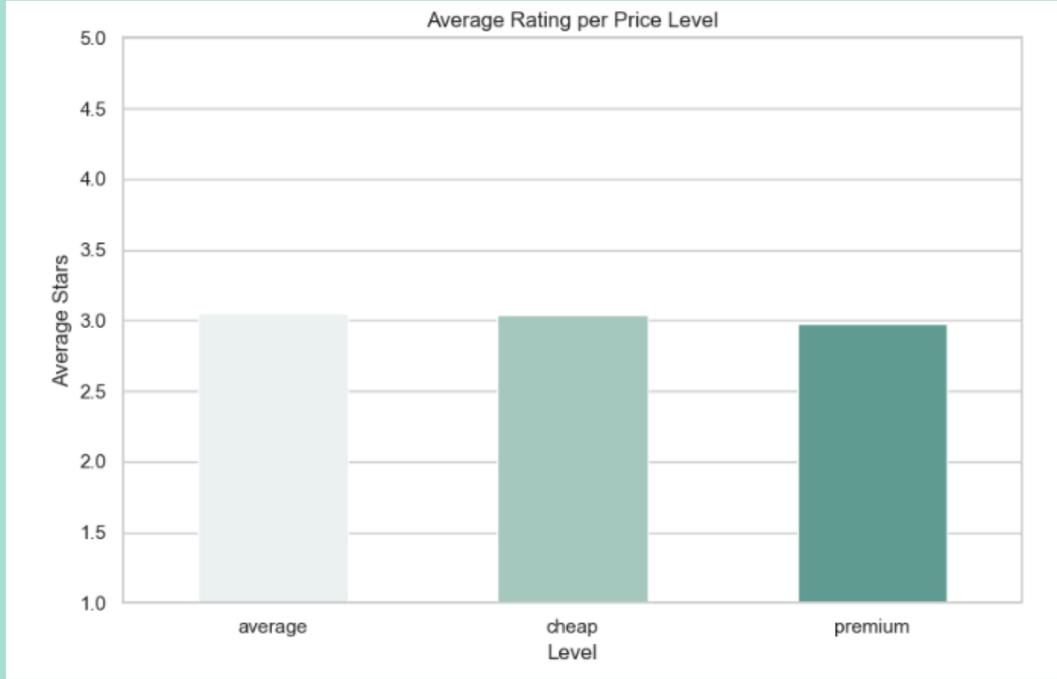
Level



X-axis: Price level

Y-axis: Item count

Premium items show the highest number of items



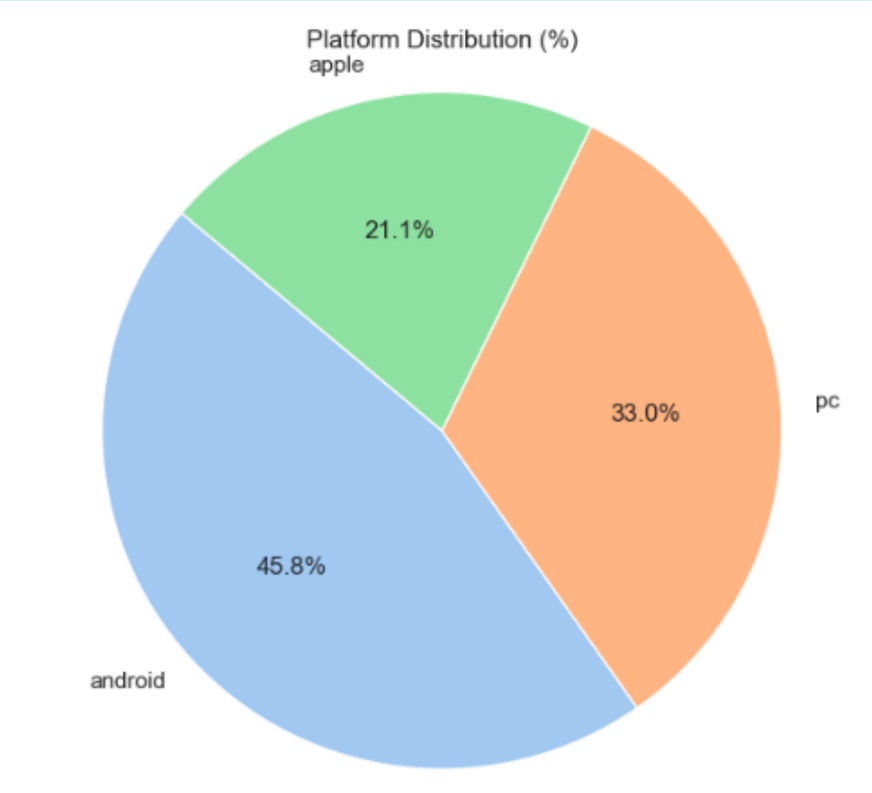
X-axis: Price level

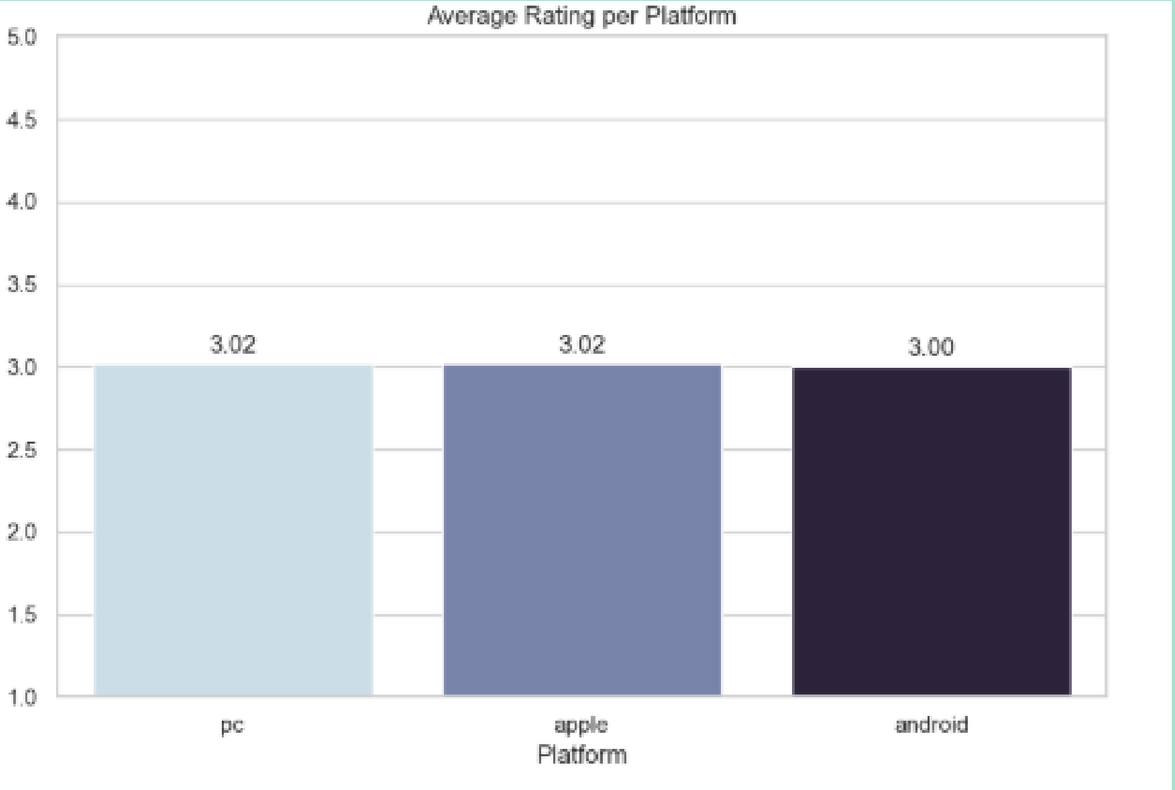
Y-axis: Average stars (rating)

The difference in ratings across the three price levels is negligible.

Visual exploration

Platform usage





With this pie chart, we can easily see that the Android platform is the most used by users.

The difference in rating is not affected by the platform used

Visual exploration - Recap 🔄

- 🔷 Volume and rating do not match perfeclty
- 💰 Price does not affect rating
- 💎 Premium level shows the highest number of exchanged items.
- 🤖 Android is the most used platform (45,8%)
  Platform does not influence rating

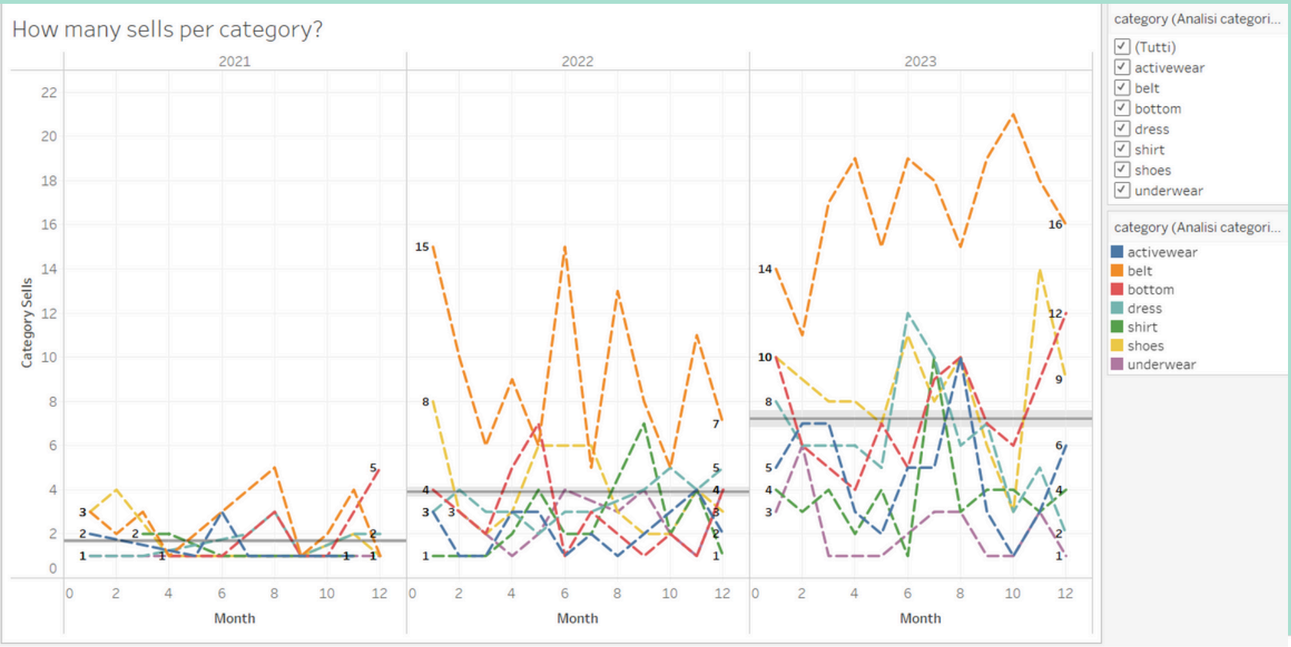
Data manipulation and processing

```sql
-- CATEGORY: how many sells per category for each month?
SELECT
    purchase_year,
    purchase_trimester,
    purchase_month,
    category,
    COUNT(*) AS category_sells
FROM
    fashion_data
GROUP BY
    purchase_year,
    purchase_trimester,
    purchase_month,
    category
ORDER BY
    purchase_year,
    purchase_trimester,
    purchase_month,
    category;
```

In this section, we focus on **sales trends by category**, using a table extracted through an SQL query that summarises sales data over time. The objective is to identify sales peaks, understand seasonal dynamics or particular events affecting demand, and detect any recurring patterns.

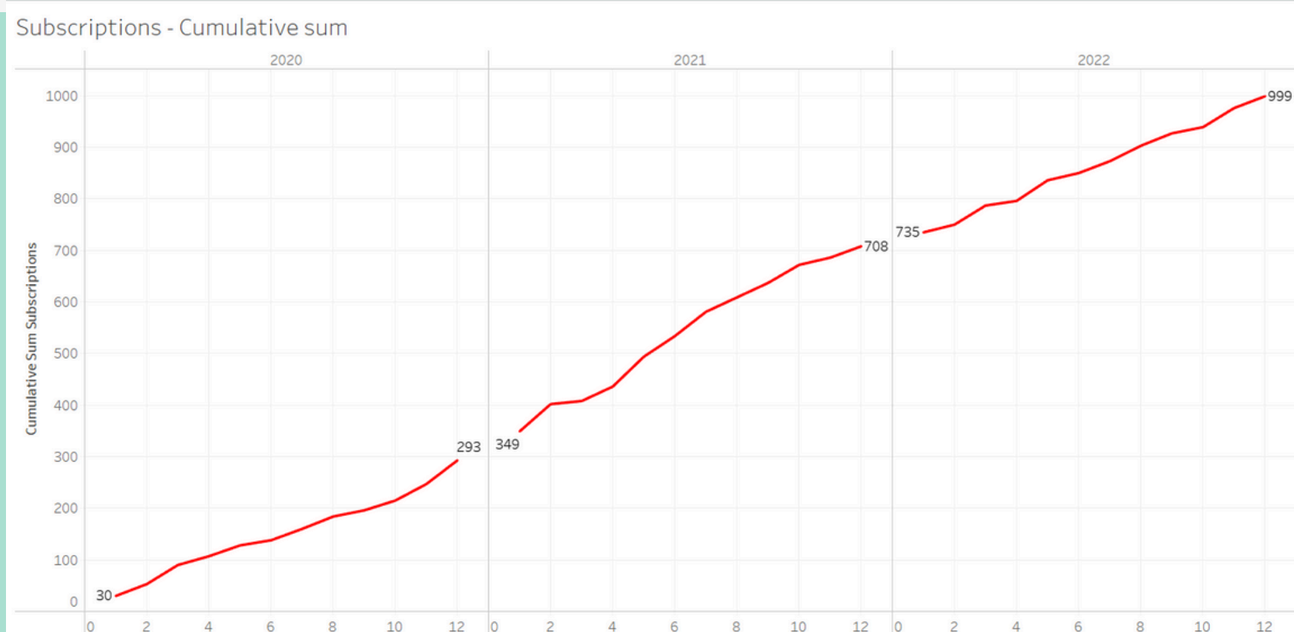
Sales trends

(Tableau)

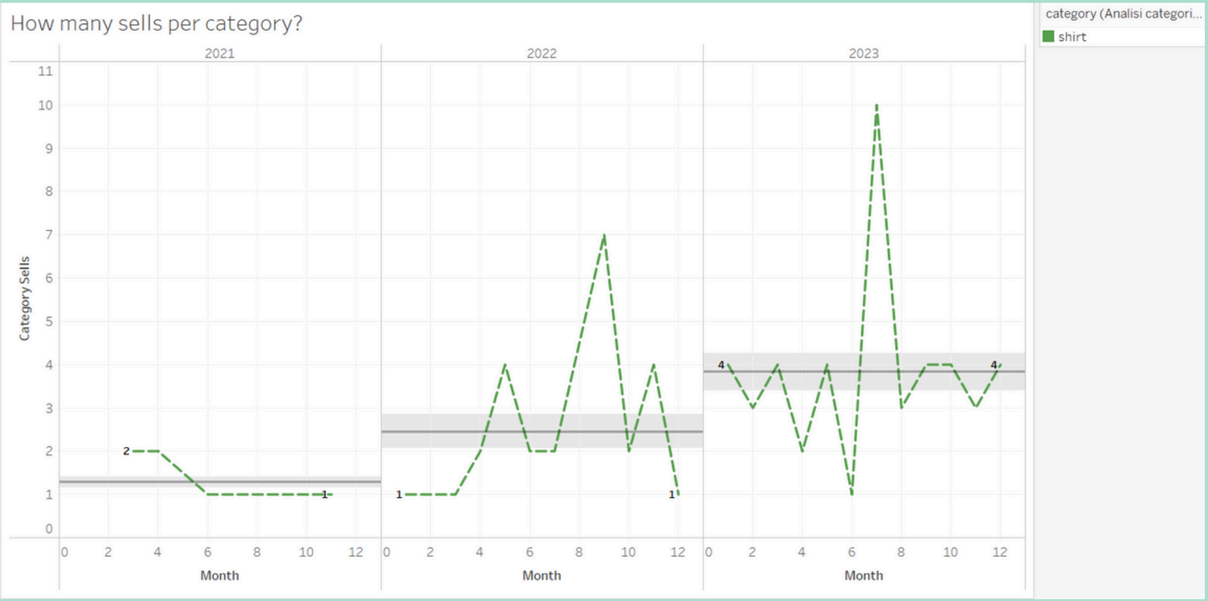


An initial analysis indicates an *avarage upward trend in category sales* over the three-year period, with peaks observed in specific months.

This growth correlates with the increasing user registrations.

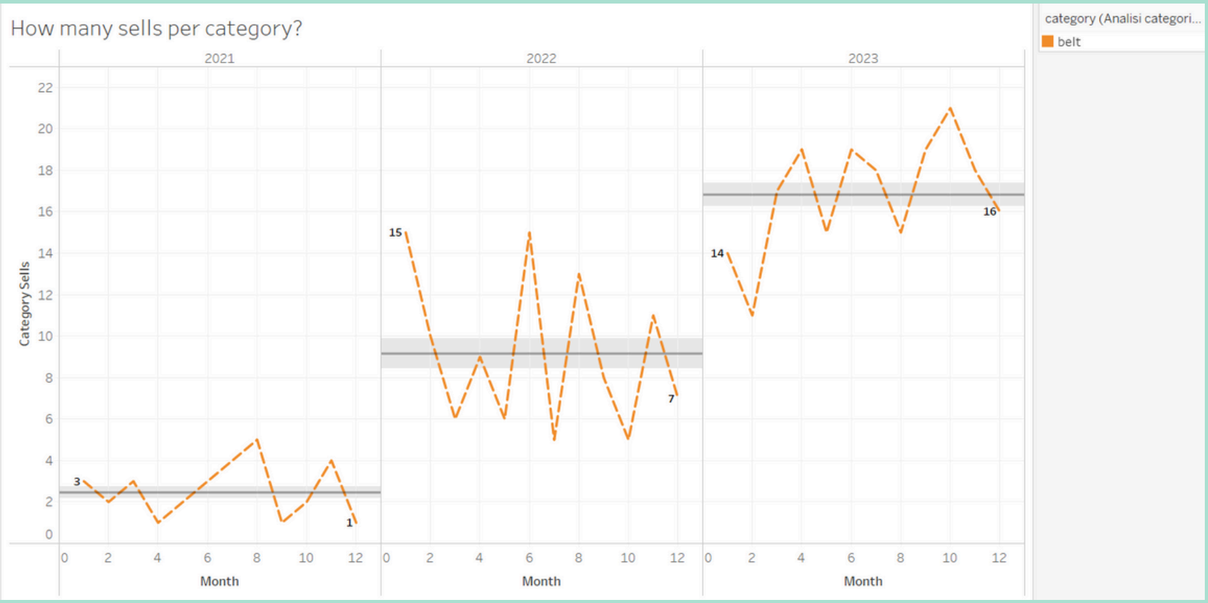


Sales trends by category



Shirts

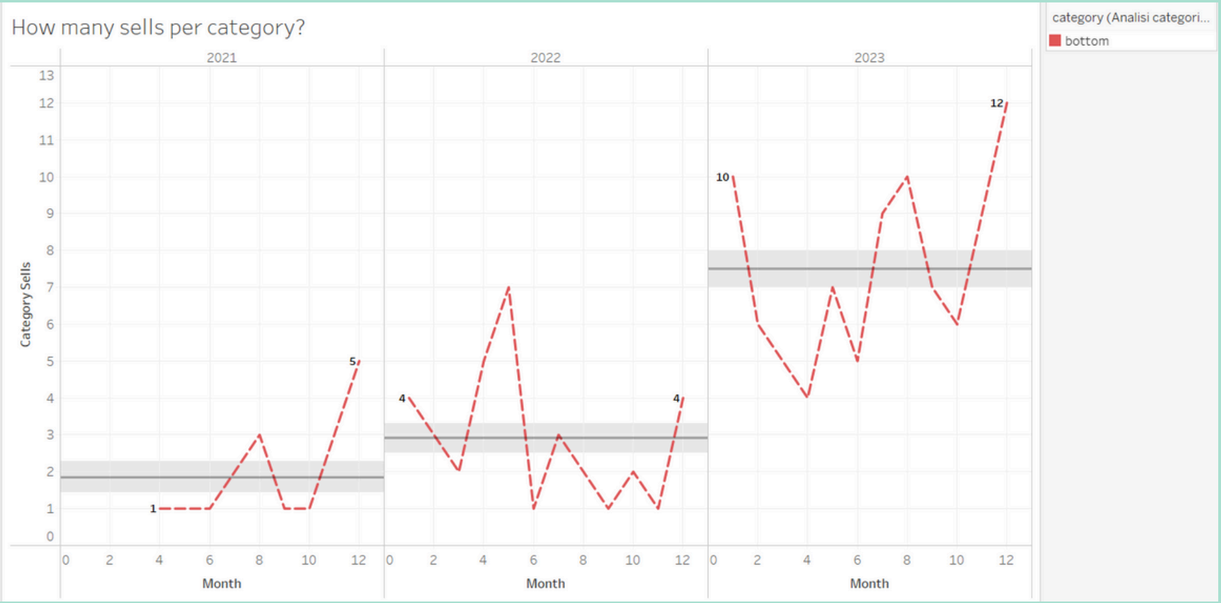
- Seasonality: Recurring sales peaks in spring and summer months (July-August)
- Interest in shirts grows in warm periods, probably for reasons of style, climatic comfort and increased social activity
- The drop in June may indicate a seasonal transition phase, fewer social events before the full summer.



Belt

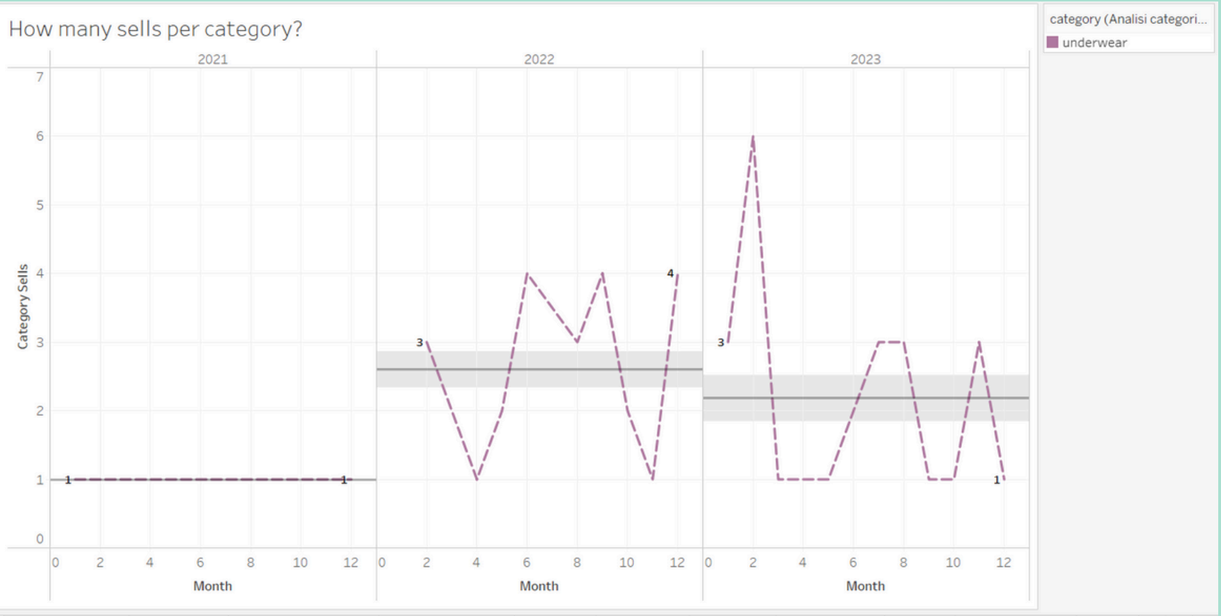
- Seasonality: April and June show recurring peaks in sales over the last two years.
- The autumn months show significant fluctuations (October and November alternation) , suggesting that specific events or promotions strongly influence demand. This indicates opportunities to optimise marketing and inventory strategies in those periods

Sales trends by category



Bottom (Trousers, Jeans, Leggings)

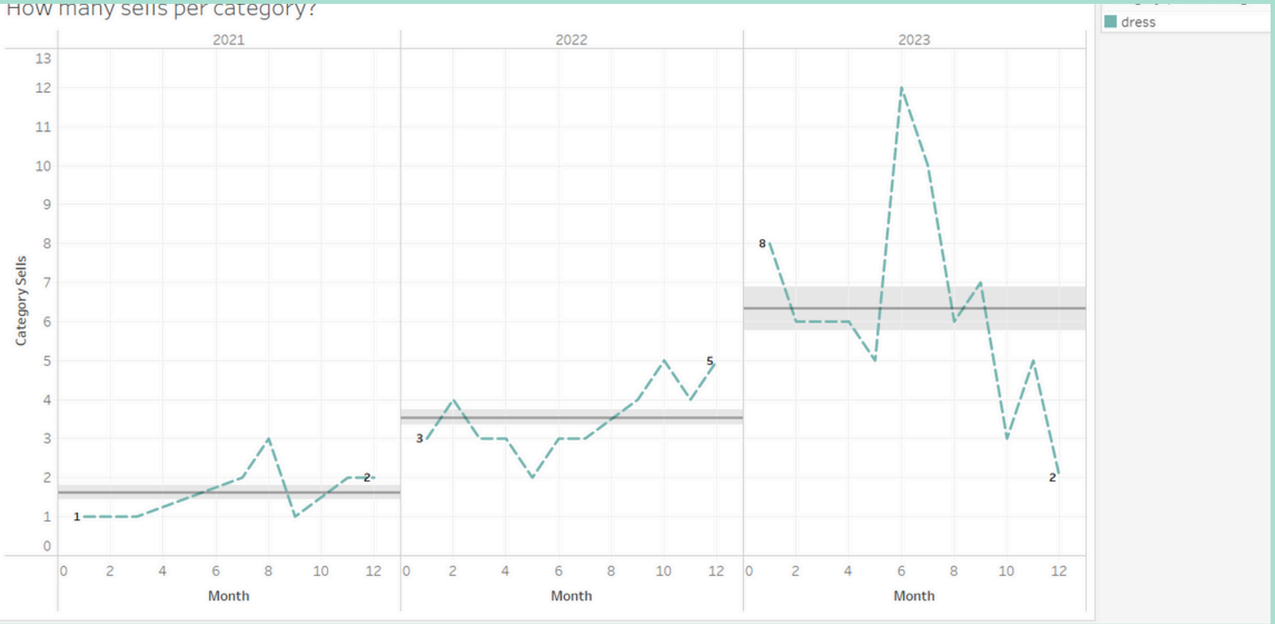
- Recurring peaks: May, August and December show annual peaks, suggesting a seasonality linked to the change of seasons and holidays.
- Plan targeted promotions for May and December, particularly on the Bottom category, to maximise sales at historical peaks.



Underwear

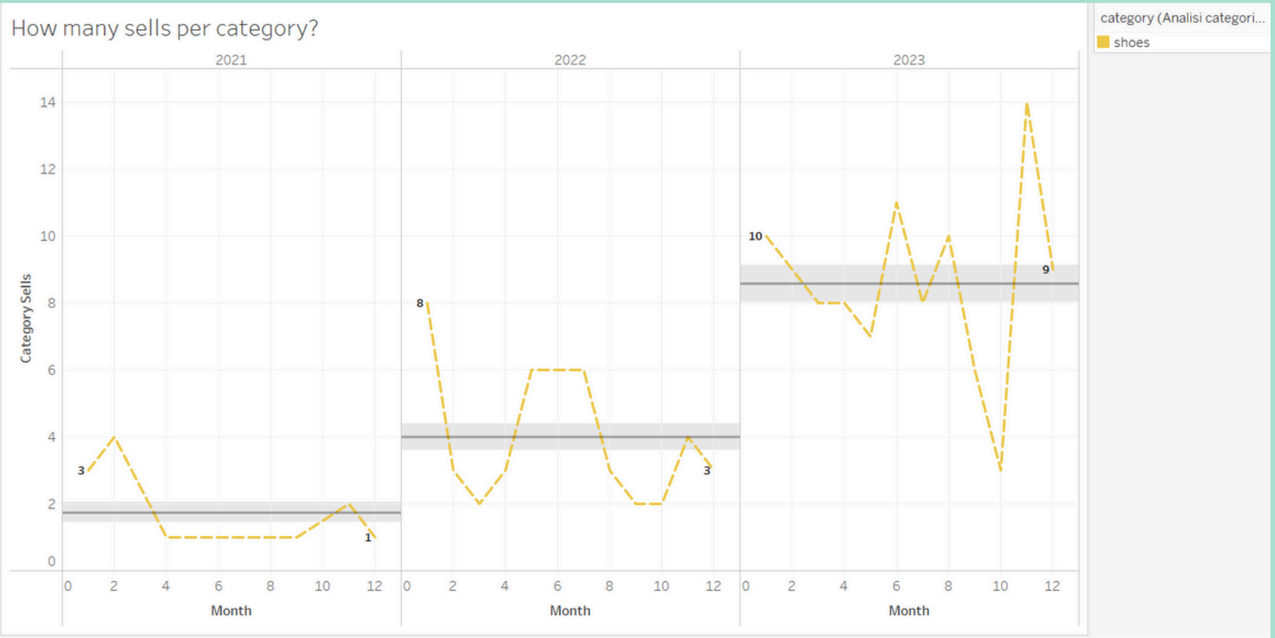
- Underwear is the only category to have decreased between 2022 and 2023
- Summer is the general peak sales period
- Doubling of sales in February 2023

Sales trends by category



Dress

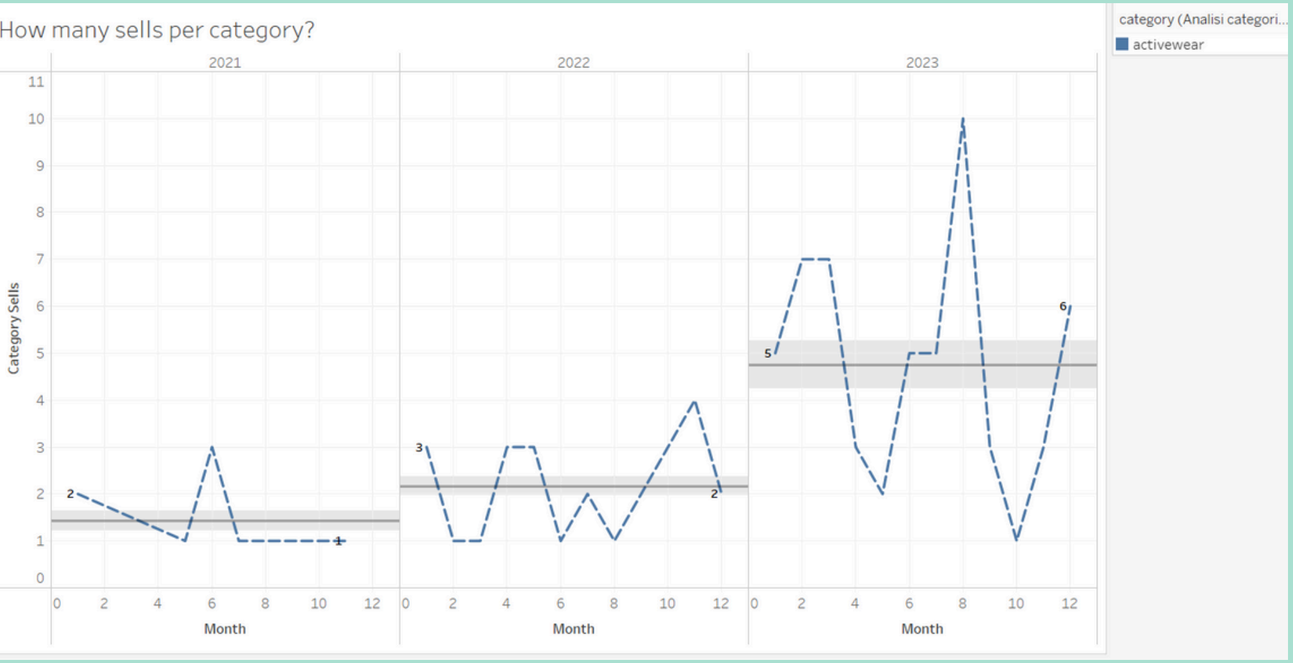
- Above-average sales since August, with clear peaks in October and December.
- In 2023, an early high peak in June, then steady decline, with volumes below average since October. Could indicate market saturation or change in tastes/habits.



Shoes

- Above-average sales in January and during the summer (June-August) in both years. Possible link with new subscription at the beginning of each year, and wardrobe summer change.
- New peak in November 2023: Possible effect of promotional events, as Black Friday, or early holiday shopping.

Sales trends by category



Activewear

- Irregular sales, with non-seasonally distributed peaks.
- Year 2022: Peaks in April-May, possible link to change of season. Second peak in November, potentially influenced by promotions or events (e.g. Black Friday).
- Year 2023: an initial peak in sales between February and March, probably related to the activity of new post-January members. This is followed by a sharp drop in spring, in stark contrast to the trend in 2022. Stable summer, sales grow rapidly, peaking in August, an anomalous behaviour that deserves further analysis. A marked slump occurs in October, possibly indicating saturation or a change in user behaviour. December shows an average recovery.

Data manipulation and modeling

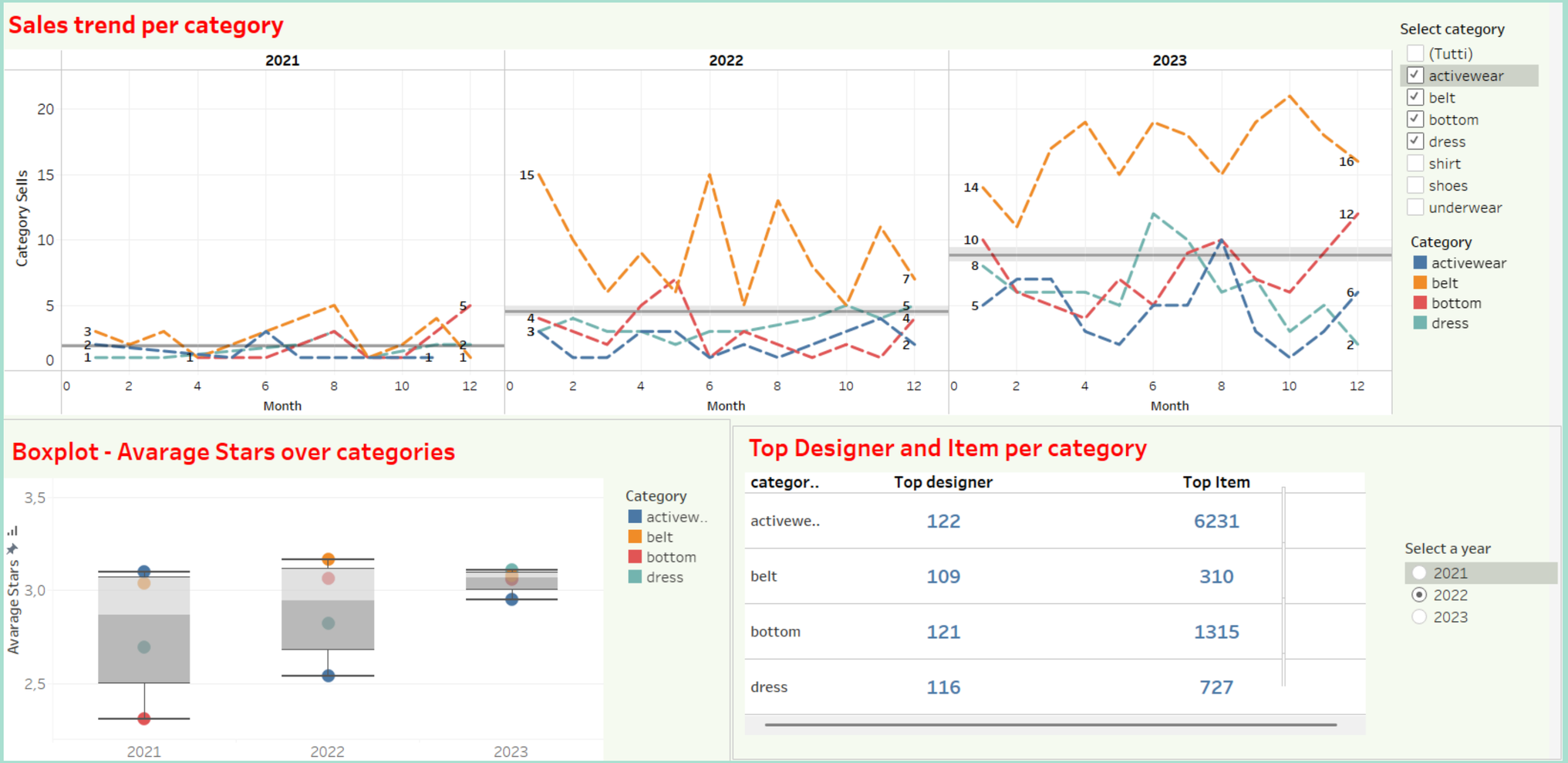
```sql
-- Top Designer / Item per category
-- Common Table Expression to calculate total sales per item, designer, category, and year
WITH top_selling_items AS (
    SELECT
        category,
        purchase_year,
        designer_id,
        item_id,
        COUNT(*) AS total_sales,

        -- Assign a row number to each item within the same category and year,
        -- ordered by number of sales in descending order
        ROW_NUMBER() OVER (
            PARTITION BY category, purchase_year
            ORDER BY COUNT(*) DESC
        ) AS rn
    FROM fashion_data
    GROUP BY category, purchase_year, designer_id, item_id
)

-- Select only the top-selling item per category and year
SELECT
    category,
    purchase_year,
    designer_id,
    item_id
FROM top_selling_items
WHERE rn = 1
ORDER BY category, purchase_year;
```

This section presents an SQL query that identifies the **top-performing designer and item — based on the highest number of sales** — for each category and year combination.

Category Analysis

Tableau Dashboard

This dashboard provides a comprehensive overview of the performance across different categories. It highlights key aspects such as temporal trends, customer ratings (via boxplots), and the top-performing designer and item for the selected year.



For a fully interactive experience, explore the dashboard on Tableau Public and dive deeper into the data by applying filters and selecting specific time periods or categories.

Market Analysis by Country

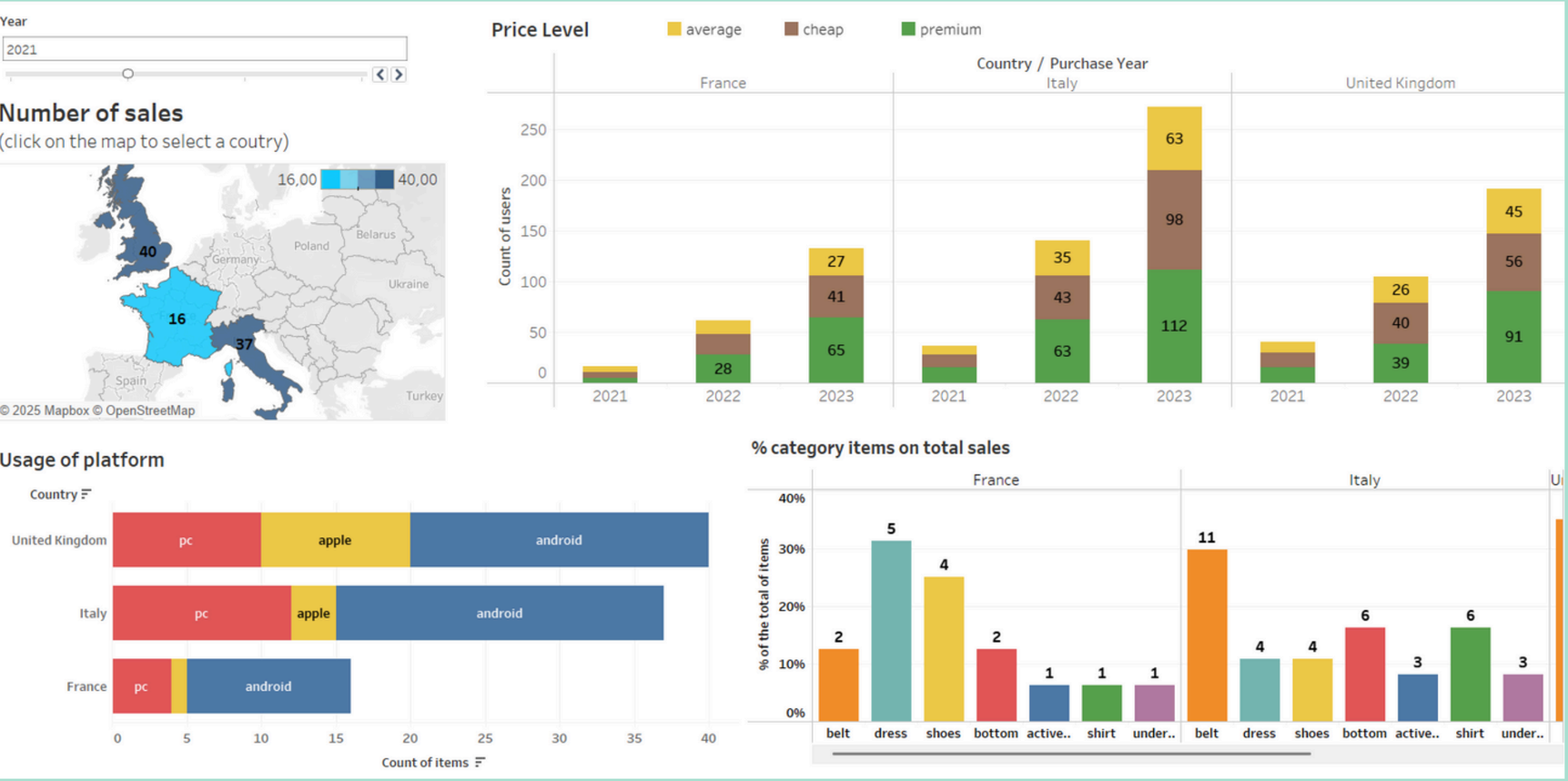
General Map

In the next section, the analysis will focus on the three countries under study: Italy, France, and the United Kingdom.

This initial map shows, through color intensity, the number of items sold in each country.
France ranks last, while Italy leads in total sales.

Market Analysis by Country

Tableau Dashboard



The dashboard combines four interactive visualizations to provide a comprehensive view of sales dynamics across different countries:

1. **Global Sales Map**: An interactive map displaying the volume of items sold per country, with a scrollable time filter to explore changes over the years.
2. **Sales by Country and Price Level:** A bar chart comparing the number of items sold in each country, broken down by price level using color shading.
3. **Platform Usage by Country:** A chart highlighting the usage level of different platforms (Apple, Android, PC) across countries.
4. **Clothing Category Distribution:** A chart showing the percentage of each clothing category relative to the total items sold, to reveal which categories dominate the market.

Market Analysis by Country

Tableau Dashboard Analysis

🌍 General Sales Trends

- As observed earlier, all three countries (Italy, France, and the UK) experienced a steady increase in total sales over the analyzed years.
- France consistently ranks last, maintaining a light blue shade on the global sales map throughout all years — a visual indicator of lower relative volume.
- Italy overtook the United Kingdom in 2021, moving from second to first in total sales — marking a strategic turning point in its market performance.

💰 Price Segment Dynamics

- The overall sales growth has been primarily driven by Premium and Cheap price segments, indicating a polarized market with strong interest at both ends of the pricing spectrum.
- The Average price range in France shows weaker growth, lagging behind other countries and segments.

💻 Platform Usage Patterns

- Android is the most commonly used platform across all countries, while Apple consistently ranks last in terms of user adoption.
- In France (2022), an exception occurs where PC and Android usage are equally represented, suggesting a unique shift in user habits or accessibility.

🧵 Category Preferences

- Each country shows distinct consumer preferences by clothing category.
- In 2021, France was an outlier: "Belt" had the lowest sales, while "Dress" had the highest, in contrast with the trends observed in Italy and the UK.
- Starting from 2022, category performance in France aligned with the broader international trend, signaling market convergence.

CLUSTERING

This SQL query performs a **behavioral clustering of users** based on two key metrics: total number of sales made and average rating received (stars). The final result is a classification of users into four groups, with names inspired by pop culture

```sql
WITH user_metrics AS (
    SELECT
        user_uuid,
        COUNT(*) AS total_sales,  -- Total number of sales per user
        ROUND(AVG(stars)::numeric, 1) AS avg_stars  -- Average rating (stars) per user, rounded to 1 decimal place
    FROM fashion_data
    GROUP BY user_uuid
),

-- Calculate median values for total_sales and avg_stars across all users
stats AS (
    SELECT
        PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY total_sales) AS median_sales,  -- Median of total sales
        PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY avg_stars) AS median_stars    -- Median of average stars
    FROM user_metrics
)
-- Assign each user to a cluster based on their total_sales and avg_stars relative to the medians
SELECT
    um.user_uuid,
    um.total_sales,
    um.avg_stars,
    CASE
        WHEN um.total_sales < s.median_sales AND um.avg_stars < s.median_stars THEN 'Jims'            -- Low sales, low rating
        WHEN um.total_sales < s.median_sales AND um.avg_stars >= s.median_stars THEN 'Mirandas'       -- Low sales, high rating
        WHEN um.total_sales >= s.median_sales AND um.avg_stars < s.median_stars THEN 'The Wolfs'      -- High sales, low rating
        WHEN um.total_sales >= s.median_sales AND um.avg_stars >= s.median_stars THEN 'The Champions'-- High sales, high rating
    END AS cluster_name
FROM user_metrics um
CROSS JOIN stats s;  -- Combine metrics with median stats for comparison
```

Users Clustering

The Jim(s)

Low sales, low ratings

Users who are not very active and whose feedback is not particularly positive.

The Wolfs

High sales, low ratings

Very active sellers, but with a perceived quality below average.

The Miranda(s)

Low sales, high ratings

They sell little, but they do it well. Users appreciate their items.

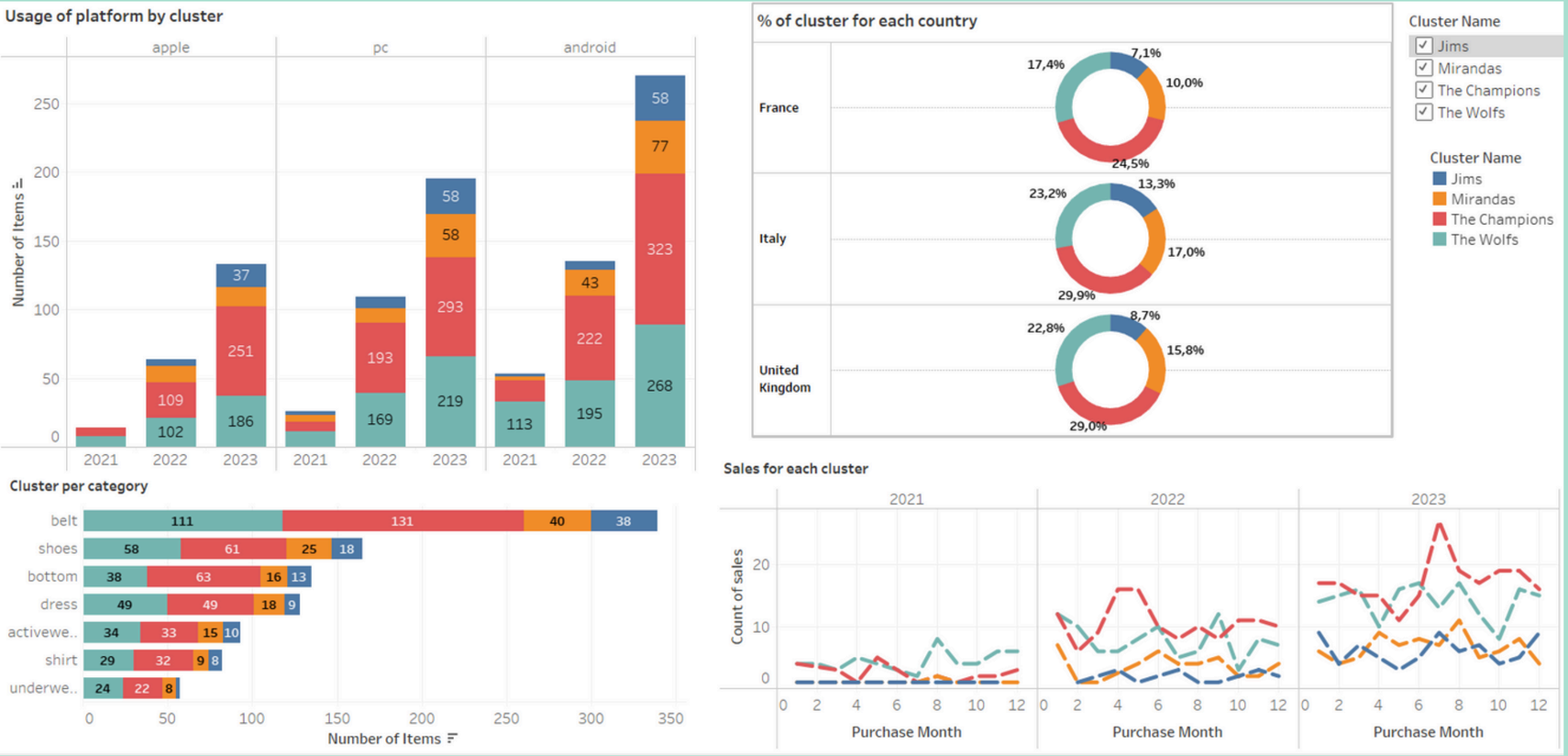
The Champions

High sales, high ratings

They sell a lot and with taste, with excellent ratings

Market Analysis by Clusters of users

Tableau Dashboard



This dashboard provides a behavioral analysis of the platform's four user clusters.

1. **Platform Usage by Cluster (Stacked Bar Chart).** It illustrates how platform usage (PC, Android, Apple) has evolved across the three-year period (2021–2023).. Each bar represents the number of transactions per year, and is broken down by cluster — allowing for a comparative view of platform preference shifts within each user group**.**
2. **Cluster Distribution by Country (Donut Chart):** it displays the average proportion of each user cluster across the main countries, highlighting how user composition varies geographically.
3. **Sales by Product Category and Cluster (Bar Chart):** a multi-colored bar chart showing the volume of items sold per product category, segmented by cluster. This allows identification of which user groups are driving sales in specific categories.
4. **Sales Trend per Cluster (Line Chart) :** it tracks the monthly sales trend for each cluster across the 2021–2023 timeframe. This view helps to detect seasonal behaviors, growth patterns, and engagement differences between clusters over time.

💡 Key Strategic Insights

1. Sustained growth driven by seasonal patterns

The average sales increase in the three-year period is closely related to the growth in user registrations.
Growth is driven mainly by the Premium and Cheap segments,

Strategy:

🎯 Plan ad hoc seasonal promotions for each category, synchronized with the identified peak moments (e.g. Dress in spring, Shoes in November)

🎯 Strengthen onboarding campaigns in the months leading up to peaks, with seasonal product suggestions for new sellers.

2. Android Platform Dominance + Cross-Cross Opportunities

Android is the most used platform in all clusters and countries, while Apple remains marginal.
The device used does not significantly impact perceived quality (reviews remain stable on all platforms).

Strategy: Multi-platform approach

🎯 Optimize experience (UX and speed) to facilitate acquisition, conversion and user loyalty, especially in high volume clusters (Wolfs, Champions).

🎯Improve navigability and desktop experience for PC platform

🎯 Launch “platform native” campaigns, such as push notifications on Android and Apple to stimulate sales at key moments.

💡 Key Strategic Insights

3. Italy leading market, France to be stimulated

Strategy:

🎯 Push localized promotions in France, adapted to price and category preferences.

🎯 Evaluate pan-European campaigns for converging categories, taking advantage of the standardization of preferences that emerged after 2022.

4. Polarized Market: Premium and Cheap Drive Growth

The mid-range has weaker performances.

Sales are more concentrated on Premium and Cheap products, but the rating remains stable even for expensive products.

Strategy:

🎯 Pushing Premium products through an higher perception of value, especially in high-rated clusters such as the Champions.

Users Clustering
Strategy

The Jims - Low-volume sellers with low ratings

Not very relevant in terms of volume, but present in core categories.

⚡ Activation potential with training tools, or quality and sales incentives

---

The Mirandas - Low-volume sellers with high ratings

Valuable segment for the platform's reputation.

⚡ Activate loyalty programs or targeted up-selling campaigns.

The Wolfs - High-volume sellers with low ratings

Clusters that are important for revenue, but can compromise the experience of other users

⚡ Activation of quality incentives, or quality control system

---

The Champions - High-volume sellers with high ratings **(Dominant category)**

Ideal cluster: quality + volume, essential for revenue

⚡ Aim for retention and loyalty through priority access to promotions or premium tools.

Conclusion

In an increasingly competitive market, selling is not enough: you need to know who sells, how and when.
This analysis shows us that behind every resold dress there is a strategy to refine, a seller to activate and an opportunity to seize.
Optimizing the experience for each cluster, guiding seasonal behaviors and customizing marketing levers is the key to transforming the platform from a simple marketplace to an intelligent engine of value.

👉 The future does not sell itself: it must be built, cluster after cluster.

GitHub

Click on this link to view the folder on GitHub containing the Notebook and SQL querie

Google Drive

Click on this link to view the folder on Drive

Tableau public

Click on this link to view the dashboards

Data Analysis final project

Presented by **Francesco Genna**