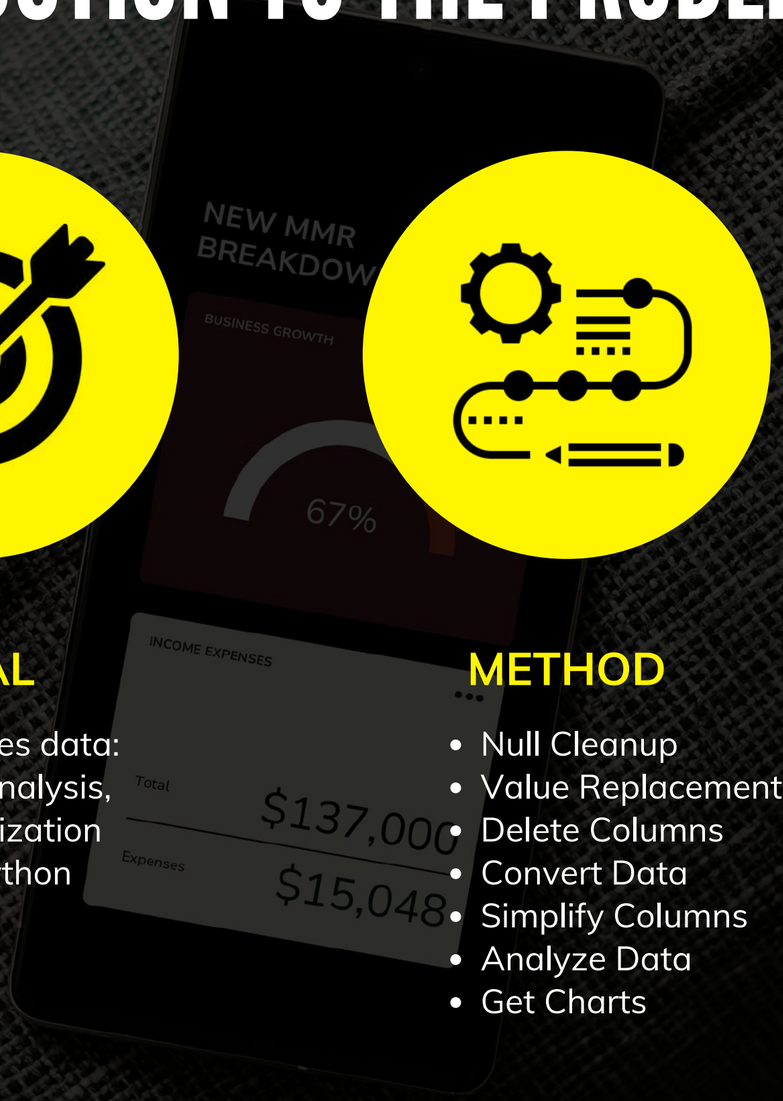
PRODUCT SALES DATA: CLEANING, ANALYSIS, AND VISUALIZATION USING PYTHON

MARÍA FERNANDA RUBÍ EGUEZ
19/08/2025

TOOLS AND LIBRARIES:
PYTHON, PANDAS, NUMPY,
POWER BI, POWER QUERY

INTRODUCTION TO THE PROBLEM



GOAL

Product sales data: cleaning, analysis, and visualization using Python

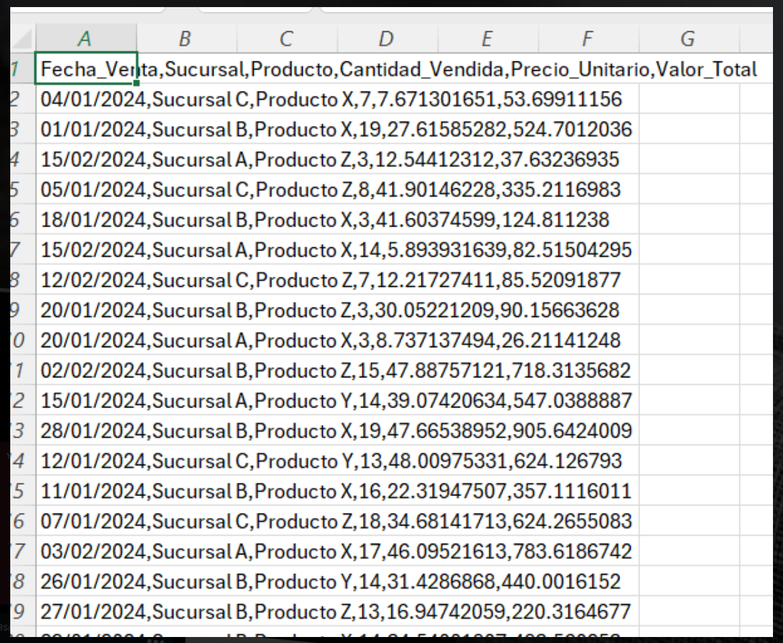
METHOD

- Null Cleanup
- Value Replacement
- Delete Columns
- Convert Data
- Simplify Columns
- Analyze Data
- Get Charts

TOOLS USED

DATA SOURCE:
PRUEBA_PYTHON.CSV

- Python (Jupyter Notebook): For analysis, modeling, and visualization.
- Pandas: For structuring and manipulating data
- Power Query: Data cleaning (removing nulls, filtering erroneous rows, ensuring consistency).
- Power BI: Visualization and analysis of the cleaned dataset.



	A	B	C	D	E	F	G
1	Fecha_Venta,Sucursal,Producto,Cantidad_Vendida,Precio_Unitario,Valor_Total						
2	04/01/2024,Sucursal C,Producto X,7,7.671301651,53.69911156						
3	01/01/2024,Sucursal B,Producto X,19,27.61585282,524.7012036						
4	15/02/2024,Sucursal A,Producto Z,3,12.54412312,37.63236935						
5	05/01/2024,Sucursal C,Producto Z,8,41.90146228,335.2116983						
6	18/01/2024,Sucursal B,Producto X,3,41.60374599,124.811238						
7	15/02/2024,Sucursal A,Producto X,14,5.893931639,82.51504295						
8	12/02/2024,Sucursal C,Producto Z,7,12.21727411,85.52091877						
9	20/01/2024,Sucursal B,Producto Z,3,30.05221209,90.15663628						
10	20/01/2024,Sucursal A,Producto X,3,8.737137494,26.21141248						
11	02/02/2024,Sucursal B,Producto Z,15,47.88757121,718.3135682						
12	15/01/2024,Sucursal A,Producto Y,14,39.07420634,547.0388887						
13	28/01/2024,Sucursal B,Producto X,19,47.66538952,905.6424009						
14	12/01/2024,Sucursal C,Producto Y,13,48.00975331,624.126793						
15	11/01/2024,Sucursal B,Producto X,16,22.31947507,357.1116011						
16	07/01/2024,Sucursal C,Producto Z,18,34.68141713,624.2655083						
17	03/02/2024,Sucursal A,Producto X,17,46.09521613,783.6186742						
18	26/01/2024,Sucursal B,Producto Y,14,31.4286868,440.0016152						
19	27/01/2024,Sucursal B,Producto Z,13,16.94742059,220.3164677						

RAW DATA

Analysis of Product Sales

```
#Import all necessary libraries
import pandas as pd
import numpy as np
```

[1] ✓ 3.9s

IMPORT LIBRARIES

```
#Upload the data
df = pd.read_csv("C:\\Users\\María Fernanda\\OneDrive\\Desktop\\MASTER\\PYTHON\\PROYECTOS\\6 Proy

#Show the firt 5 rows of the table
df.head(5)
```

[2] ✓ 0.0s

UPLOAD THE FILE

CHECK FOR NULL VALUES

No null values are found in the data frame.

RENAME FROM SPANISH TO ENGLISH

The entire list was translated, both in columns and rows, from Spanish to English using rename.

INDUSTRY BACKGROUND

```
#Clean the null values
df.isnull().sum().sum()
```

✓ 0.0s

```
np.int64(0)
```

```
#Renaming the columns from spanish to english
df.rename(columns={'Fecha_Venta':'Date_sale'},inplace=True)
df.rename(columns={'Sucursal':'Branch'},inplace=True)
df.rename(columns={'Producto':'Product'},inplace=True)
df.rename(columns={'Cantidad_Vendida':'Quantity_sold'},inplace=True)
df.rename(columns={'Precio_Unitario':'Unit_price'},inplace=True)
df.rename(columns={'Valor_Total':'Total_value'},inplace=True)

df = df.replace({"Sucursal": "Branch"}, regex=True)
df = df.replace({"Producto": "Product"}, regex=True)

df.head(5)
```

✓ 0.0s

	Date_sale	Branch	Product	Quantity_sold	Unit_price	Total_value
0	04/01/2024	Branch C	Product X	7	7.671302	53.699112
1	01/01/2024	Branch B	Product X	19	27.615853	524.701204
2	15/02/2024	Branch A	Product Z	3	12.544123	37.632369
3	05/01/2024	Branch C	Product Z	8	41.901462	335.211698

It was transformed into the Date_sale column to Datetime format so that it could be used in the analysis later.

1

```
#Transforming the Date_sale column into the DD/MM/YYYY date format.
df['Date_sale'] = pd.to_datetime(df['Date_sale'], format='%d/%m/%Y')
df['Date_sale'].dtype
```

```
dtype('<M8[ns]')
```

Once the date was obtained, they were ordered in ascending order to have a better order.

2

```
#Order the dates ascending
df = df.sort_values(by="Date_sale", ascending=True)
df.head(5)
```

	Date_sale	Branch	Product	Quantity_sold	Unit_price	Total_value
99	2024-01-01	Branch C	Product X	17	35.516568	603.781657
1	2024-01-01	Branch B	Product X	19	27.615853	524.701204
24	2024-01-03	Branch A	Product Y	17	41.412326	704.009541
98	2024-01-04	Branch B	Product Z	14	46.607772	652.508803
0	2024-01-04	Branch C	Product X	7	7.671302	53.699112

Finally, from the Date_sale column, a new column called month was created so that we could then work with that column to perform different analyses.

3

```
#New column (month per year)
df['Month_year'] = df['Date_sale'].dt.strftime('%m/%Y')
df[['Date_sale', 'Month_year']].head(5)
```

✓ 0.0s

	Date_sale	Month_year
99	2024-01-01	01/2024
1	2024-01-01	01/2024
24	2024-01-03	01/2024
98	2024-01-04	01/2024
0	2024-01-04	01/2024

DATA ANALYSIS

Within the code, sales were analyzed by product, brand and month, both in value and quantity.

```
#EXPLORATORY ANALYSIS (EDA)
#Total sales by product/branch/month
sales_per_branch = df.groupby("Branch")["Total_value"].sum().reset_index()
sales_per_product = df.groupby("Product")["Total_value"].sum().reset_index()
sales_per_month = df.groupby("Month_year")["Total_value"].sum().reset_index()

#Total quantity by product/branch/month
quantity_per_month = df.groupby("Month_year")["Quantity_sold"].sum().reset_index()
quantity_per_product = df.groupby("Product")["Quantity_sold"].sum().reset_index()
quantity_per_Branch = df.groupby("Branch")["Quantity_sold"].sum().reset_index()

display(sales_per_branch)
display(sales_per_month)
display(sales_per_product)
display(quantity_per_Branch)
display(quantity_per_month)
display(quantity_per_product)
```

0.0s

	Branch	Quantity_sold
0	Branch A	284
1	Branch B	475
2	Branch C	336

	Month_year	Quantity_sold
0	01/2024	659
1	02/2024	436

	Product	Quantity_sold
0	Product X	484
1	Product Y	335
2	Product Z	276

	Branch	Total_value
0	Branch A	7192.487110
1	Branch B	15501.593026
2	Branch C	10693.715321

	Month_year	Total_value
0	01/2024	20052.551667
1	02/2024	13335.243790

	Product	Total_value
0	Product X	14818.844393
1	Product Y	10313.002304
2	Product Z	8255.948759

DATA ANALYSIS

In this code, what was done was to analyze what the maximum values were for each of the categories in both branch, product, and month.

```
#Show the answers
print(f"The branch with the highest total sales value is: {top_branch_sales} with a total of {value_top_branch:,.2f}")
print(f"The product with the highest total sales value is: {top_product_sales} with a total of {value_top_product:,.2f}")
print(f"The month with the highest total sales value is: {top_month_sales} with a total of {value_top_month:,.2f}")
print(f"The branch with the highest total quantity is: {top_branch_quantity} with a total of {value_top_branch2:,.2f}")
print(f"The product with the highest total quantity is: {top_product_quantity} with a total of {value_top_product2:,.2f}")
print(f"The month with the highest total quantity is: {top_month_quantity} with a total of {value_top_month2:,.2f}")
```

```
#Find the row with the maximum value
#TOP Sales per branch
top_row1 = sales_per_branch.loc[sales_per_branch['Total_value'].idxmax()]
top_branch_sales = top_row1['Branch']
value_top_branch = top_row1['Total_value']

#TOP Sales per product
top_row2 = sales_per_product.loc[sales_per_product['Total_value'].idxmax()]
top_product_sales = top_row2['Product']
value_top_product = top_row2['Total_value']

#TOP Sales per month
top_row3 = sales_per_month.loc[sales_per_month['Total_value'].idxmax()]
top_month_sales = top_row3['Month_year']
value_top_month = top_row3['Total_value']

#TOP quantities per branch
top_row4 = quantity_per_Branch.loc[quantity_per_Branch['Quantity_sold'].id
top_branch_quantity = top_row4['Branch']
value_top_branch2 = top_row4['Quantity_sold']

#TOP quantities per product
top_row5 = quantity_per_product.loc[quantity_per_product['Quantity_sold'].i
top_product_quantity = top_row5['Product']
value_top_product2 = top_row5['Quantity_sold']

#TOP quantities per month
top_row6 = quantity_per_month.loc[quantity_per_month['Quantity_sold'].idxmax()]
top_month_quantity = top_row6['Month_year']
value_top_month2 = top_row6['Quantity_sold']
```

```
The branch with the highest total sales value is: Branch B with a total of 15,501.59
```

```
The product with the highest total sales value is: Product X with a total of 14,818.84
```

```
The month with the highest total sales value is: 01/2024 with a total of 20,052.55
```

```
The branch with the highest total quantity is: Branch B with a total of 475.00
```

```
The product with the highest total quantity is: Product X with a total of 484.00
```

```
The month with the highest total quantity is: 01/2024 with a total of 659.00
```

DATA ANALYSIS

This code is used to calculate the average unit price of products and branches

In addition, the average amount sold is also calculated using the .mean() method.

```
#Average prices and totals

#Average price of products
avg_price_prod = df.groupby("Product")["Unit_price"].mean().reset_index()

#Average price of branch
avg_price_bra = df.groupby("Branch")["Unit_price"].mean().reset_index()

#Average quantity sold
avg_sold_quan = df["Quantity_sold"].mean()

display(avg_price_bra)
display(avg_price_prod)
display(avg_sold_quan)
```

	Branch	Unit_price
0	Branch A	24.510105
1	Branch B	31.532713
2	Branch C	29.536179

	Product	Unit_price
0	Product X	30.561207
1	Product Y	28.974270
2	Product Z	26.548544

The average quantity sold is 10 units

```
np.float64(10.95)
```

DATA ANALYSIS

Within the code, sales were analyzed by product, brand and month, both in value and quantity.

The following tables show the percentage contribution of each branch to total sales. Branch B stands out with the highest share at 46.42%. In terms of products, Product X achieved the largest monetary sales, representing 44.38% of the total.

```
#Percentage share of total sales per branch
total_sales = sales_per_branch["Total_value"].sum()
sales_per_branch ["Participation_%"] = (sales_per_branch["Total_value"] / total_sales) * 100

#Percentage share of total sales per product
total_sales2 = sales_per_product['Total_value'].sum()
sales_per_product['Participation_%'] = (sales_per_product['Total_value'] / total_sales2) * 100

display(sales_per_branch)
display(sales_per_product)
```

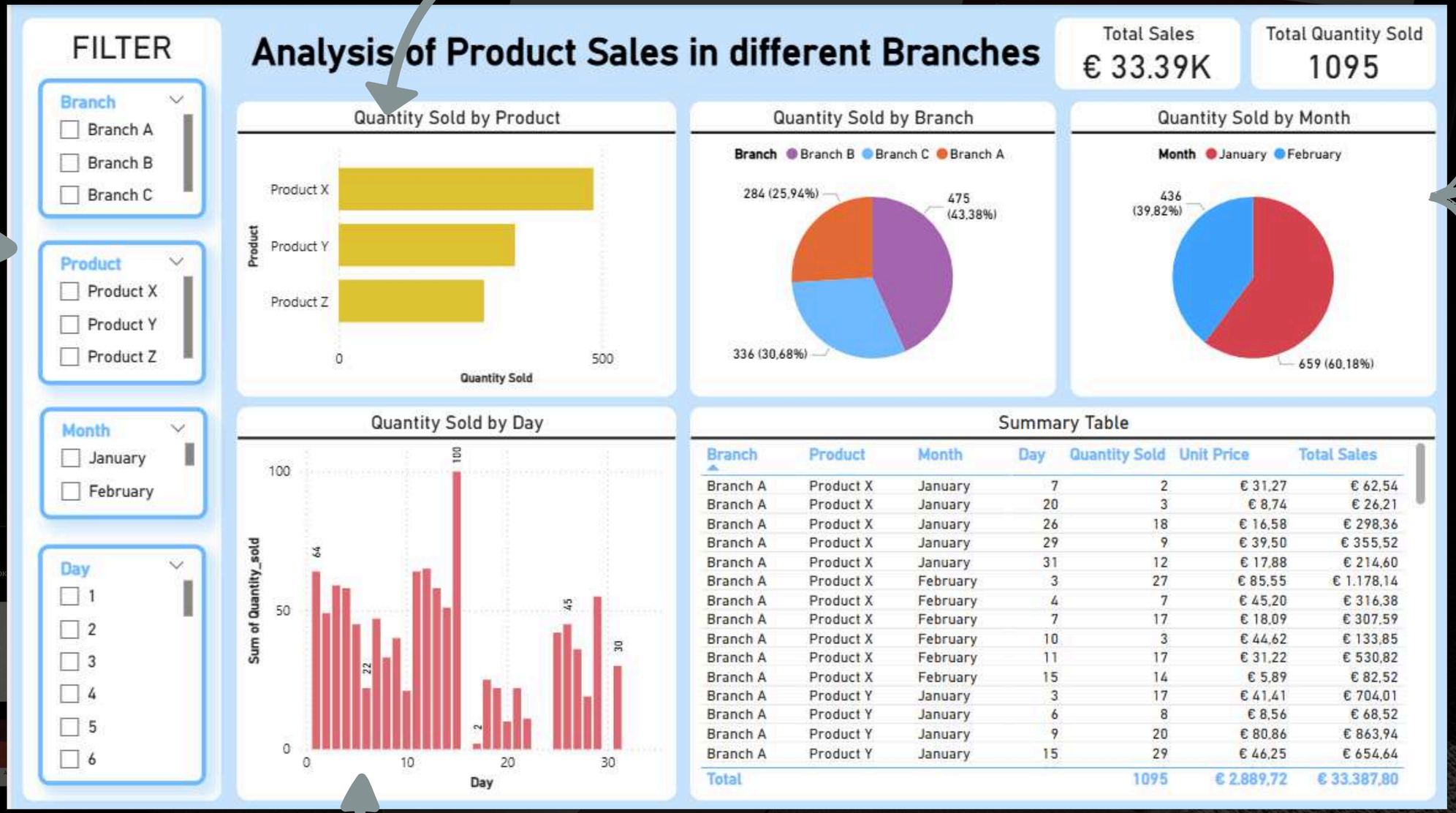
✓ 0.0s

	Branch	Total_value	Participation_%
0	Branch A	7192.487110	21.542264
1	Branch B	15501.593026	46.428921
2	Branch C	10693.715321	32.028815

	Product	Total_value	Participation_%
0	Product X	14818.844393	44.384016
1	Product Y	10313.002304	30.888539
2	Product Z	8255.948759	24.727445

DATA ANALYSIS
INSIGHTS THE DASHBOARD PROVIDES:

Product X is the best-selling of the 3, based on the quantity sold



Branch	Product	Month	Day	Quantity Sold	Unit Price	Total Sales
Branch A	Product X	January	7	2	€ 31,27	€ 62,54
Branch A	Product X	January	20	3	€ 8,74	€ 26,21
Branch A	Product X	January	26	18	€ 16,58	€ 298,36
Branch A	Product X	January	29	9	€ 39,50	€ 355,52
Branch A	Product X	January	31	12	€ 17,88	€ 214,60
Branch A	Product X	February	3	27	€ 85,55	€ 1.178,14
Branch A	Product X	February	4	7	€ 45,20	€ 316,38
Branch A	Product X	February	7	17	€ 18,09	€ 307,59
Branch A	Product X	February	10	3	€ 44,62	€ 133,85
Branch A	Product X	February	11	17	€ 31,22	€ 530,82
Branch A	Product X	February	15	14	€ 5,89	€ 82,52
Branch A	Product Y	January	3	17	€ 41,41	€ 704,01
Branch A	Product Y	January	6	8	€ 8,56	€ 68,52
Branch A	Product Y	January	9	20	€ 80,86	€ 863,94
Branch A	Product Y	January	15	29	€ 46,25	€ 654,64
Total				1095	€ 2.889,72	€ 33.387,80

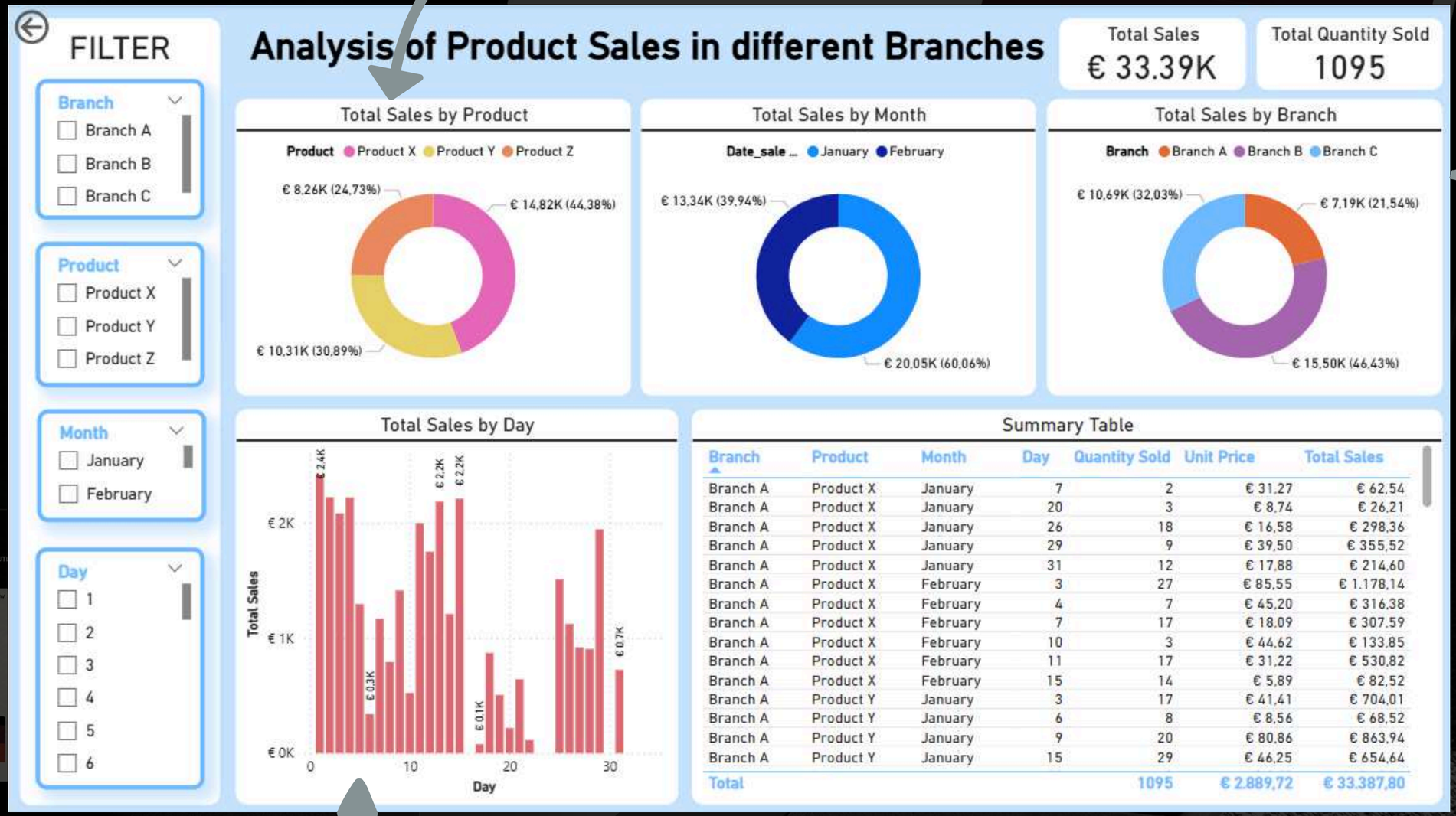
The filters section allows you to customize the view based on your needs. You'll find it on each dashboard sheet.

According to this pie chart, January was the best month with the most units sold, with a total of 659, representing 60.18% of the total.

During the first 15 days of the month, unit sales were strong, peaking on the 15th, which stands out as the best day of the month.

DATA ANALYSIS
INSIGHTS THE DASHBOARD PROVIDES:

Product X is the best-selling of the 3, based on total sales.



Branch	Product	Month	Day	Quantity Sold	Unit Price	Total Sales
Branch A	Product X	January	7	2	€ 31,27	€ 62,54
Branch A	Product X	January	20	3	€ 8,74	€ 26,21
Branch A	Product X	January	26	18	€ 16,58	€ 298,36
Branch A	Product X	January	29	9	€ 39,50	€ 355,52
Branch A	Product X	January	31	12	€ 17,88	€ 214,60
Branch A	Product X	February	3	27	€ 85,55	€ 1.178,14
Branch A	Product X	February	4	7	€ 45,20	€ 316,38
Branch A	Product X	February	7	17	€ 18,09	€ 307,59
Branch A	Product X	February	10	3	€ 44,62	€ 133,85
Branch A	Product X	February	11	17	€ 31,22	€ 530,82
Branch A	Product X	February	15	14	€ 5,89	€ 82,52
Branch A	Product Y	January	3	17	€ 41,41	€ 704,01
Branch A	Product Y	January	6	8	€ 8,56	€ 68,52
Branch A	Product Y	January	9	20	€ 80,86	€ 863,94
Branch A	Product Y	January	15	29	€ 46,25	€ 654,64
Total				1095	€ 2.889,72	€ 33.387,80

Based on the graph, Branch B recorded the highest total sales, reaching a value of 15.50k, which represents 46.43% of the overall sales among the three active branches

The first half of the month accounts for the majority of sales, with revenue peaks occurring between the 1st and 5th, and again between the 10th and 15th.

I WANT TO SAY

THANK YOU

FOR YOUR ATTENTION