Retail Analytics in shopping malls

Analytical project by Nataliia Shevchenko

March 2024

Data sources

- **Customer Shopping Dataset - Retail Sales Data** - https://www.kaggle.com/datasets/mehmettahiraslan/customer-shopping-dataset - Exploring Market Basket Analysis in Istanbul Retail Data. 100k records of retail sales activity in 8 shopping malls in Istanbul in 2021, 2022 and 2023
- **OpenExchangeRates API** - https://docs.openexchangerates.org - historical currency exchange rates of Turkish lira to US dollar. Used to account for inflation in Turkey in 2021, 2022 and 2023
- **Visual Crossing Weather API** - https://www.visualcrossing.com - historical weather in Istanbul in 2021, 2022 and 2023

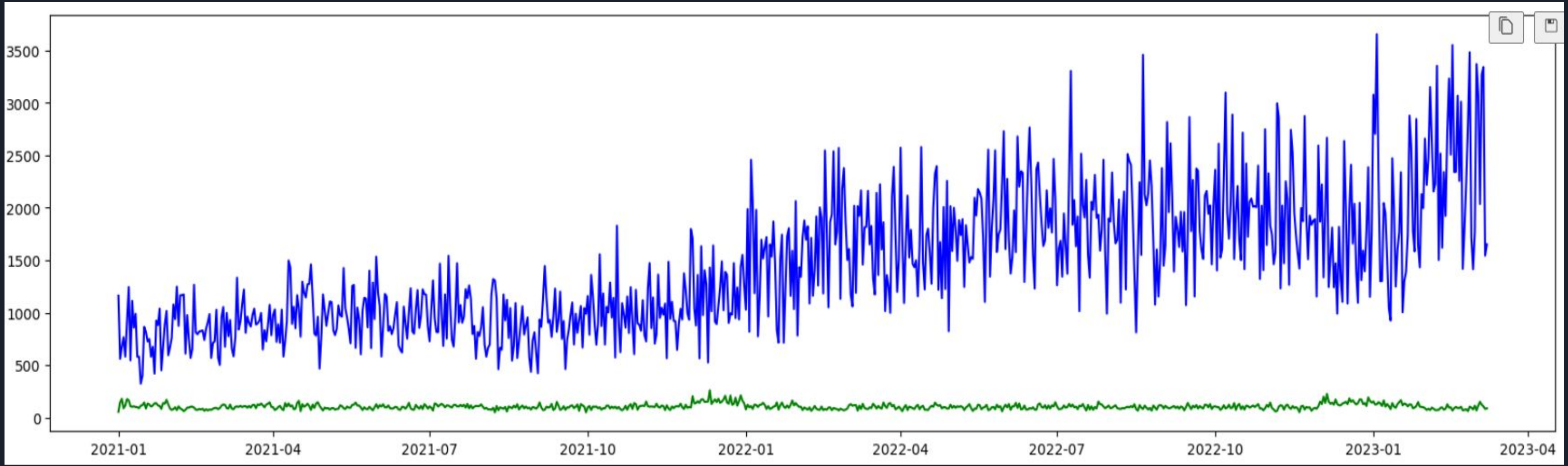
Project structure

Sources	Kaggle	OpenExchangeRates API	Visual Crossing Weather API
Ingestion	**Manual download** Customer shopping dataset *Resources\source\customer_shopping_data.csv*	**CurrencyExchangeAPI.ipynb** Ingestion of historical currency exchange rates *Resources\output\exchange_rate.csv*	**WeatherAPI.ipynb** Ingestion of historical weather *Resources\output\Istanbul_historical_weather.csv*
Processing	**DataDiscovery.ipynb** Initial data discovery on raw shopping data	**DataPreparation.ipynb** Merging customer shopping dataset with historical currency exchange rates and weather; calculating cost metrics in USD; adding age buckets and calendar columns; renaming and reordering *Resources\output\customer_shopping_data.csv*	
Analysis	**DataAnalytics.ipynb** Common dependencies; Shopping malls traffic; Dependency on weather conditions; Average weather conditions over year; Seasonal changes of average price and total cost for different categories		

Technology stack:

1. Python: pandas, matplotlib, scipy.stats, requests
2. Jupyter Notebook (IronPython), REST API, CSV

Data Discovery



Main observations:

1. There are only 100k records to analyze, each record has unique **invoice_no** and unique **customer_id**, so there are no opportunities for neither basket analysis nor customer behavior over time analysis
2. Date is presented from 1/1/2021 till 3/8/2023, total 797 days
3. Total quantity per day shows seasonal peaks, while average daily price shows big inflation and seasonal peaks
4. There is 60:40 ratio for women's and men's transactions for a whole data set
5. There are only 8 categories of products and only 10 shopping malls to analyze

Currency Exchange API

Key features:

1. Using a loop across unique days only to limit a number of API requests
2. Date format conversion from DD/MM/YYYY to YYYY-MM-DD to meet API format
3. Saving result to CSV file

```python
#Set the API base URL
base_url = "https://openexchangerates.org/api/historical"

date = []
exchange_rate = []

#Loop through all dates. Currently, a limit is set to the first 3 dates to avoid using up the free A
#It's better to conduct experiments on 2-3 dates to test the loop and ensure that the API returns va
#For a production launch, use for invoice_date in df:

for invoice_date in df:

    #Assemble the final string for the API with all parameters.
    url = f"{base_url}/{invoice_date}.json?app_id={openexchangerates_api_key}&base=USD&symbols=TRY"

    #Implement logging that is convenient for visual monitoring, as the full run takes 7-10 minutes.
    print(f"currently processing {invoice_date}...")

    # Make the API request
    response = requests.get(url)

    # Convert response to JSON
    data = response.json()
    date.append(invoice_date)
    exchange_rate.append(data["rates"]["TRY"])
```

Weather API

Key features:

1. Using a loop across unique monthly ranges only to limit a number of API requests
2. Additional parsing of the response to retrieve daily data
3. Additional parsing of lists to have a fully flat final structure
4. Date format conversion from DD/MM/YYYY to YYYY-MM-DD to meet API format
5. Saving result to CSV file

```python
#Start a loop over the DataFrame containing date ranges. For testing purposes, head(3) is used.
for index, row in dates_df.head(3).iterrows():  #for index, row in dates_df.iterrows():
    start_date = row['first_date_of_month'].date()
    end_date = row['last_date_of_month'].date()

    url = f"{base_url}/{city_name}/{start_date}/{end_date}?unitGroup=us&include=days&key={visualcrossing_api_key}&contentType=json"

    response = requests.get(url)

    #Convert response to JSON
    data = response.json()

    #Since the API returns a set of dates (1 month), a loop is needed for each day to extract the data.
    #Perform a loop over the days list.

    for d in data["days"]:
        date.append(d["datetime"])
        tempmax.append(d["tempmax"])
        tempmin.append(d["tempmin"])
        temp.append(d["temp"])
        feelslikemax.append(d["feelslikemax"])
        feelslikemin.append(d["feelslikemin"])
        feelslike.append(d["feelslike"])
        dew.append(d["dew"])
        humidity.append(d["humidity"])
        precip.append(d["precip"])
        precipprob.append(d["precipprob"])
        precipcover.append(d["precipcover"])

       #Since 'preciptype' is returned either as a null value (None) or as a list,
        #for example, ["rain"] or ["rain", "snow"],
        #the purpose of this code is to extract all values from the list (if it is not empty) and list them separated by commas,
        #for instance, ["rain"] -> "rain", ["rain", "snow"] -> "rain, snow"
        #The goal is to avoid storing a list in the final dataset.
```

Data Preparation

Key features:

1. Merge with exchange rates to consider inflation by converting to USD
2. Merge with historical weather in Istanbul
3. Calculating two types of age buckets
4. Calculating calendar columns for seasonal analytics and year-over-year comparison
5. Renaming and reordering
6. Saving result to CSV file

```
#Apply currency conversion to create a new column with price in USD,
#dividing the price in TYR by the exchange rate (dividing because we have downloaded the exchange rate of USD to TYR)

customer_shopping_data_df["Price (USD)"] = customer_shopping_data_df["Price (TYR)"] / customer_shopping_data_df["Exchange Rate (USD-to-TYR)"]

#Calculate the cost in lira, based on the price and quantity of the purchased product

customer_shopping_data_df["Cost (TYR)"] = customer_shopping_data_df["Price (TYR)"] * customer_shopping_data_df["Quantity"]
customer_shopping_data_df["Cost (USD)"] = customer_shopping_data_df["Price (USD)"] * customer_shopping_data_df["Quantity"]
customer_shopping_data_df.head()
```

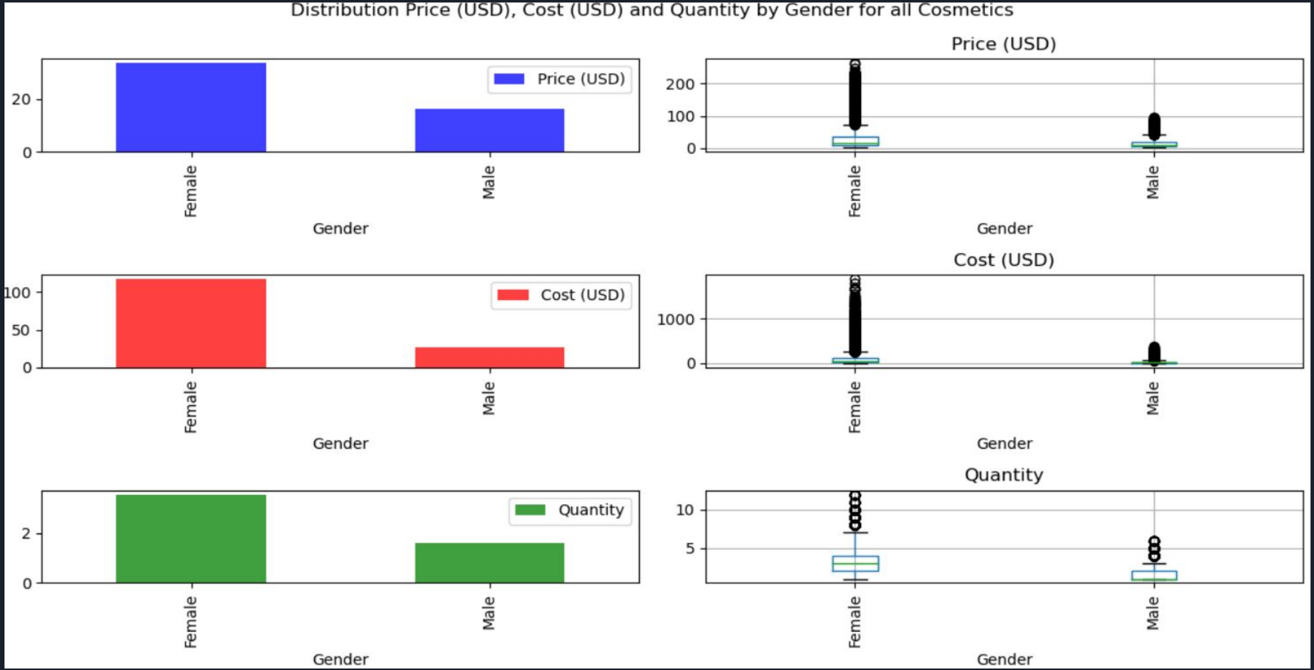
Invoice #	Customer ID	Gender	Age	Category	Quantity	Price (TYR)	Payment Method	Invoice Date	Shopping Mall	...	UV Index	Sunrise	Sunset	Conditions	Description
I138884	C241288	Female	28	Clothing	4	16107.36	Credit Card	2022-08-05	Kanyon	...	NaN	06:03:31	20:16:03	Partially cloudy	Partly cloudy throughout the day.
I317333	C111565	Male	21	Shoes	3	220.54	Debit Card	2021-12-12	Forum Istanbul	...	2.0	08:19:55	17:35:41	Rain, Partially cloudy	Partly cloudy throughout the day with rain.
I127801	C266599	Male	20	Clothing	1	180.43	Cash	2021-11-09	Metrocity	...	3.0	07:44:31	17:50:44	Rain, Partially cloudy	Partly cloudy throughout the day with late aft...
I173702	C988172	Female	66	Shoes	5	848.38	Credit Card	2021-05-16	Metropol AVM	...	8.0	05:45:11	20:16:18	Rain, Partially cloudy	Partly cloudy throughout the day with rain.

Data Analytics

Key features:

1. Utilizing a prepared CSV file minimizes additional data transformations, enhancing performance and reducing memory usage. Selecting only necessary columns from the dataset further optimizes memory consumption
2. Applying the following statistical tests:
   a. ANOVA test is applied for comparing metrics distribution across different attribute values, replacing T-tests for two groups
   b. Chi-square test is employed to analyze mall traffic distribution by gender
   c. Correlation coefficients are calculated for time-series analysis
3. Custom data visualization functions, namely **metrics_distribution_by_attribute** and **time_series_plots**, are utilized to simplify development and prioritize data analysis
4. The **pivot_table()** method of DataFrame is utilized to compute year-over-year weather conditions distribution, with stacked bar charts used to identify months with specific weather conditions
5. Analysis encompasses various combinations of product categories and attributes, with focus on the most relevant findings included in the final analysis file

Data Analytics. Common Analysis



Cosmetics by Gender:

p-value = 1.637e-200
There are statistically significant differences in the distribution of Price (USD) by Gender.

p-value = 0.0
There are statistically significant differences in the distribution of Cost (USD) by Gender.

p-value = 0.0
There are statistically significant differences in the distribution of Quantity by Gender.

The most interesting common dependencies found:

1. On average, women purchase more cosmetics than men across price, cost, and quantity metrics.
2. Within the Clothing category, individuals aged 35-50 tend to buy more expensive products, while maintaining similar purchase quantities.
3. On average, women buy more souvenirs than men, with prices remaining consistent between the two genders

Data Analytics. Shopping mall traffic

Female traffic

Shopping Mall	observed	expected
Cevahir AVM	2940	2984.954925
Emaar Square Mall	2842	2877.302774
Forum Istanbul	3016	2958.639955
Istinye Park	5874	5849.698282
Kanyon	11906	11855.492183
Mall of Istanbul	11902	11927.260283
Metrocity	8941	8977.591341
Metropol AVM	6144	6076.963934
Viaport Outlet	2949	2938.903727
Zorlu Center	2968	3035.192596

Male traffic

Shopping Mall	observed	expected
Cevahir AVM	2051	2006.045075
Emaar Square Mall	1969	1933.697226
Forum Istanbul	1931	1988.360045
Istinye Park	3907	3931.301718
Kanyon	7917	7967.507817
Mall of Istanbul	8041	8015.739717
Metrocity	6070	6033.408659
Metropol AVM	4017	4084.036066
Viaport Outlet	1965	1975.096273
Zorlu Center	2107	2039.807404

Shopping mall traffic by gender:

Female traffic

Critical value = 16.918
statistic=5.002, pvalue=0.834
p-value > 0.05, we cannot reject H0; therefore, the difference in women's shopping mall traffic is random.

Male traffic

Critical value = 16.918
statistic=7.444, pvalue=0.591
p-value > 0.05, we cannot reject H0; therefore, the difference in men's shopping mall traffic is random.

Analysis:

1. Utilizing the common 60:40 ratio of women to men, expected traffic can be calculated for shopping malls
2. Chi-square test helps identify if there are preferences among women or men for specific shopping malls
3. The analysis indicates that neither gender exhibits a preference for particular shopping malls, suggesting equal patronage across locations

Data Analytics. Dependency On Weather Conditions

Shoes by Precip Type:

p-value = 0.866
There are no statistically significant differences in the distribution of Price (USD) by Precip Type.

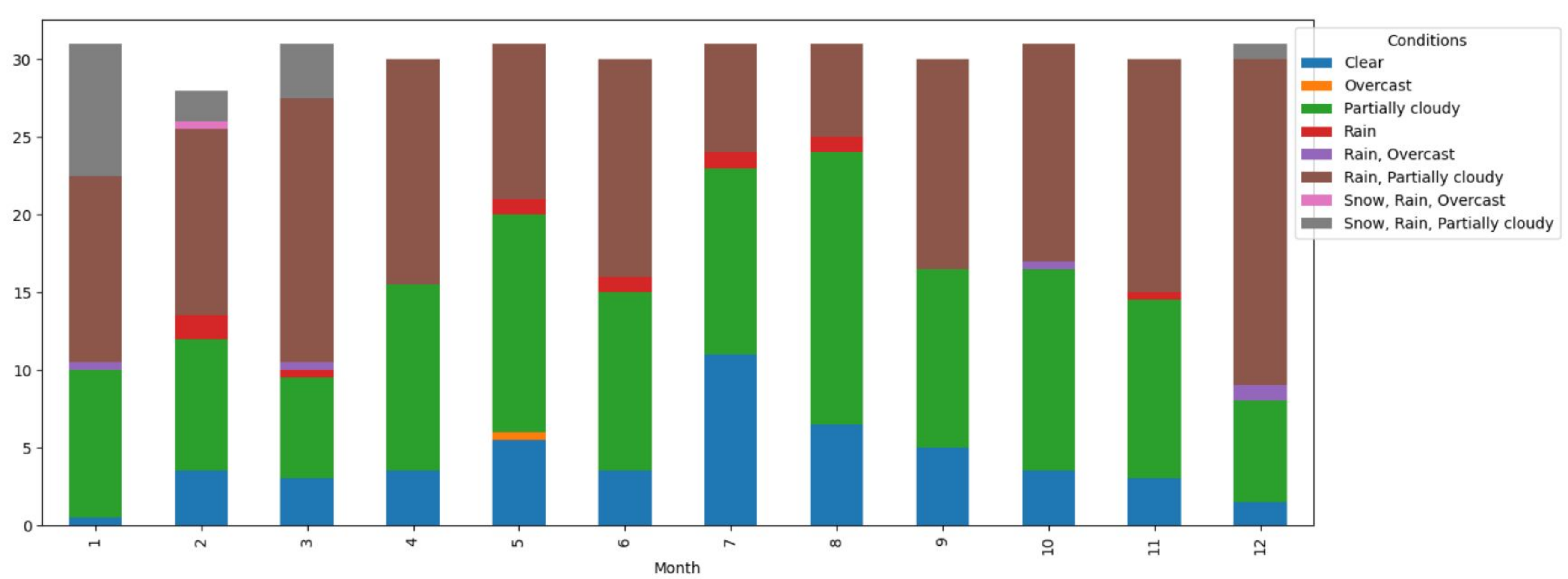
p-value = 0.143
There are no statistically significant differences in the distribution of Cost (USD) by Precip Type.

p-value = 0.035
There are statistically significant differences in the distribution of Quantity by Precip Type.

Analysis:

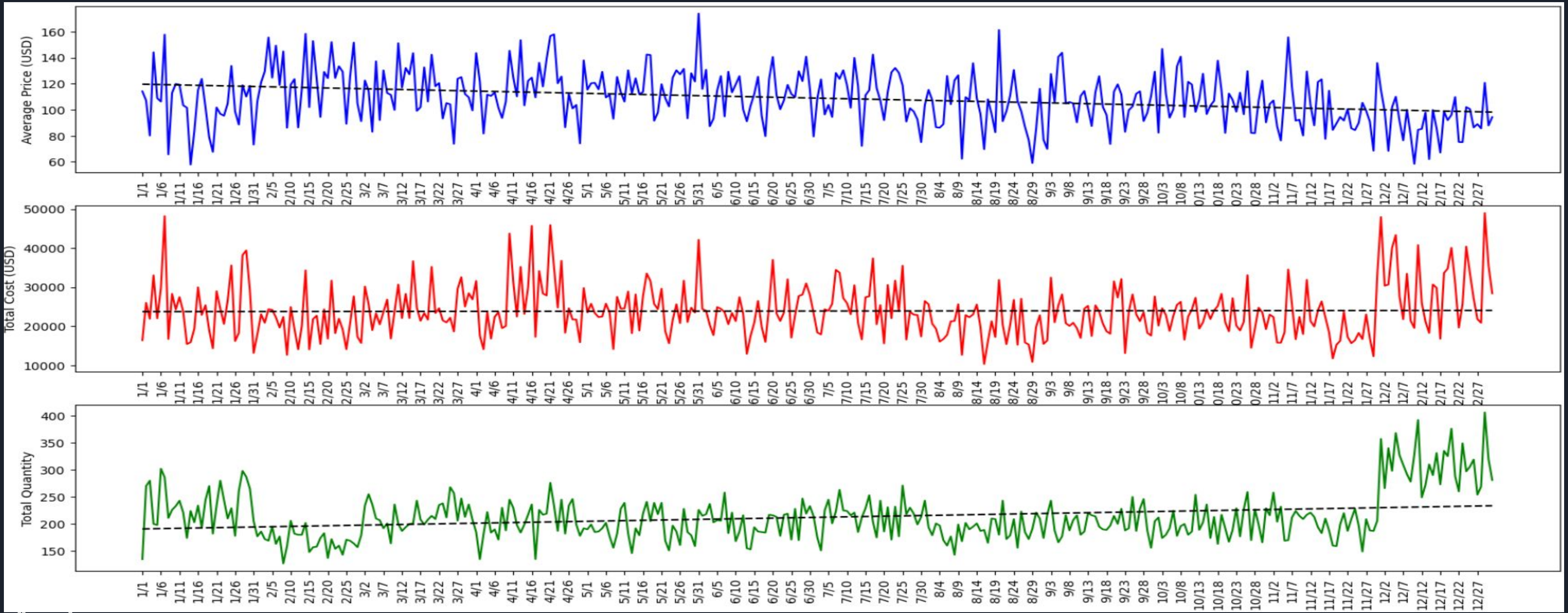
1. Shoe sales typically increase on rainy or snowy days compared to days with no precipitation.
2. This trend is confirmed distribution of price, cost, and quantity by weather conditions and having snow days
3. This suggests a higher demand for winter footwear during rain/snow weather, prompting quicker purchases

Data Analytics. Weather Conditions By Month



Analysis:
Snow and rain weather conditions are primarily experienced during January, February, and March, with occasional occurrences in December as well. Consequently, this presents an opportunity for manufacturers and shopping malls to strategically plan marketing campaigns for shoe sales during these months, capitalizing on consumer needs driven by inclement weather..

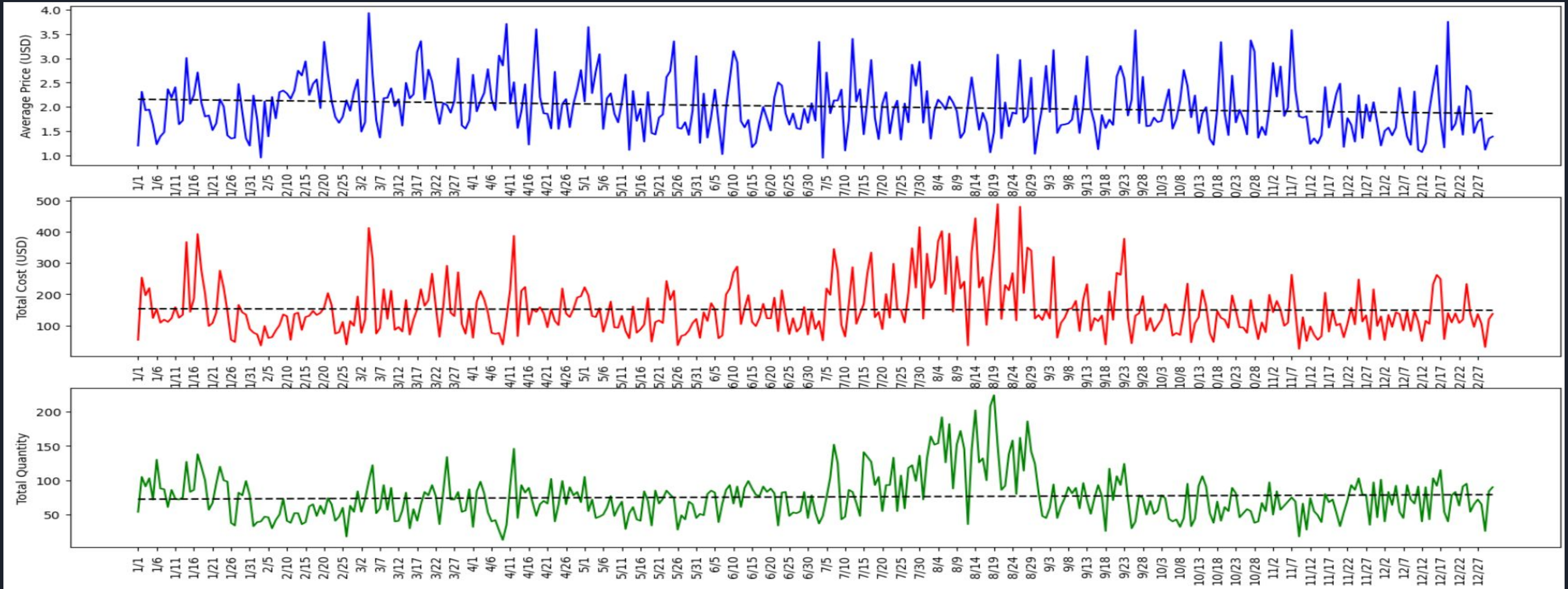
Data Analytics. Seasonal Changes For Clothing



Analysis:

1. Towards the end of the year, prices show a tendency to decrease, potentially indicating the influence of the December sales season
2. Additionally, notable peaks in both total daily cost and quantity occur in December. This phenomenon suggests the effectiveness of marketing strategies implemented during this period

Data Analytics. Seasonal Changes For Souvenirs



Analysis:

1. Price exhibits no discernible trends throughout the year
2. However, there are noticeable peaks in both total daily cost and quantity during July and August, particularly pronounced in August. This observation suggests a seasonal trend, likely attributable to increased tourism during these months

Conclusions

Analytical insights:

1. When dealing with foreign currencies, it's crucial to account for inflation. Converting to USD is a common approach to address this concern
2. Remaining open to various hypotheses about the data and investigating any discovered dependencies is essential for validation and uncovering underlying reasons
3. In searching for seasonal trends, it is advisable to analyze complete years of data to prevent biased results

Key Takeaways:

1. Stage data ingested from APIs to avoid excessive API calls. Limiting API calls, especially to unique dates, is vital for both restricted and paid pricing plans. Whenever possible, retrieve data once and reuse it across multiple analyses to optimize resource usage
2. Separate data ingestion and data transformations from data discovery and analytics to maintain logical separation and streamline team development processes
3. Utilize functions for repetitive tasks to enhance analytical performance. By implementing complex functions, we can reduce the amount of code needed to call them. In this project, only two lines of code were required to invoke functions that generate charts and conduct statistical analyses for multiple metrics simultaneously. This approach enables efficient analysis across numerous combinations, allowing focus on identified dependencies

Questions